SYSTEMTHEORETISCHE UNTERSUCHUNG DES RECHNERUNTERSTÜTZTEN LEHRPROGRAMMIERENS

VON

KLAUS-DIETER GRAF

1971
VERLAG SCHNELLE QUICKBORN

GRUNDLAGENSTUDIEN

AUS

KYBERNETIK

UND GEISTESWISSENSCHAFT

BEIHEFT ZU BAND 12

1971

Verlag Schnelle, Eberhard und Wolfgang Schnelle GmbH, Quickborn

Druck und Einband: Maurischat & Bevensee, Quickborn
Printed in Germany

SYSTEMTHEORETISCHE UNTERSUCHUNG DES RECHNERUNTERSTÜTZTEN LEHRPROGRAMMIERENS

von Klaus-Dieter Graf

SYSTEMTHEORETISCHE UNTERSUCHUNG DES RECHNERUNTERSTÜTZTEN LEHRPROGRAMMIERENS

Einleitung:
Die Objektivierung einiger Lehrerfunktionen durch rechnerunterstützte Programmierung (RUP) und rechnerunterstützte Unterweisung (RUU).

Elektronische Datenverarbeitungsanlagen sind in den vergangenen Jahren von vielen Seiten als Werkzeuge bei der Erstellung und Darbietung von Lehrprogrammen herangezogen worden. Einen Überblick für den deutschsprachigen Bereich gibt Lehnert 1970. Dort wird die Bedeutung von elektronischer Datenverarbeitung in Schule und Ausbildung insgesamt angesprochen; wir beschränken uns hier nur auf den Sektor der Lehrprogramme. Zur generellen Problematik äußert sich Fischer 1971.

In dem uns interessierenden Sektor trennt man i.a. R U P: rechnerunterstützte (Lehr)Programmierung, und R U U: rechnerunterstützte Unterweisung. Angesichts der Komplexität der in diesem Bereich erforderlichen Vollzüge vorwiegend geisteswissenschaftlicher Art, deren Kalkülisierbarkeit oder Algorithmierbarkeit bislang nur in sehr geringem Umfange untersucht wurde, beschränkte man sich zunächst - zumindest bei den bereits praktisch eingesetzten Verfahren - auf die Objektivierung einiger weniger Lehrerfunktionen innerhalb der Erstellung und Darbietung von Unterweisung, insbesondere also von Lehrprogrammen. Als Lehrerfunktionen bezeichnen wir hier solche Tätigkeiten, die im Falle personalen Unterrichts i.a. vom Lehrer im Verlaufe von Vorbereitung und Abwicklung des Unterrichts zu vollziehen sind. Im übrigen orientieren wir uns terminologisch am Lexikon der kybernetischen Pädagogik und der programmierten Instruktion (Englert u.a. 1966).

Beim Lehrprogrammieren mit Hilfe von Autorsprachen wie COURSEWRITER oder LIDIA (Müller u. Wolber 1970, Stobbe 1970) wird z.B. der Rechner zur Entgegennahme und Protokollierung von Lehrschritten sowie zu deren Einfügung in die Makrostruktur eines Lehralgorithmus eingesetzt, wohingegen Aufgaben wie die Lehrstoff- und Lehrzielfestlegung oder die didaktisch geeignete Gestaltung der Lehrschritte personal vom Autor gelöst werden.

Die Leistungen des Rechners bei der Darbietung von Lehrprogrammen, z.B. im Rahmen von C A L: computer assisted learning (Hartley und Putschkat 1970), simulieren i.a. Lehrerfunktionen wie das Anbieten von Lehrschritten über geeignete Peripheriegeräte, die Entgegennahme von Adressatenreaktionen (meist aus-

gewählt aus einem kleinen, vorgegebenen Repertoire) und die Auswahl eines neuen Lehrschritts aufgrund dieser Reaktionen anhand eines vorher vollständig festgelegten Lehralgorithmus. Daneben werden Aufgaben statisischer und verwaltungsmäßiger Art wahrgenommen (Unterrichtskontrolle).

Die Beantwortung einer vom Adressaten frei formulierten Anfrage kann dagegen vom Rechner ebensowenig als Lehrerfunktion vollzogen werden wie die Beurteilung und Verwertung einer frei gewählten Antwort als Adressatenreaktion.

Dennoch ist das Spektrum bereits heute objektivierbarer Lehrerfunktionen im Bereich von R U P und R U U wesentlich größer, als es die praktischen Anwendungen vermuten lassen. Allerdings ist zur Realisierung dieser Objektivationen z. T. ein viel höherer Aufwand an Peripheriegeräten und an Programmierung erforderlich. (Programmierung hier als Rechnerprogrammierung verstanden.)

Hinsichtlich des R U P zeigen z. B. Formaldidaktiken, daß es grundsätzlich möglich ist, echt didaktische Vollzüge bei der Makrostruktur- und auch Mikrostrukturgestaltung algorithmisch abwickeln zu lassen. (Frank 1966, 1967, Frank und Graf 1967, Blischke u. a. 1968, Frank 1969a, Arlt 1970, Graf 1969). Verteilungsalgorithmen optimieren den Begriffsaufbau und -fortschritt in Lehrprogrammen (Lánský 1970, 1971a), Dialog-Didaktik integriert personale Vollzüge und algorithmische Beurteilung derselben sowie algorithmische konstruktive Vorschläge (Graf 1969a, 1970).

Bezüglich der vorbereitenden Aufgaben ermöglicht ANEX (Analyse von Explanationen) eine Lehrstoff- und Lehrzielerfassung im Dialog zwischen Autor und Rechner, bei der Vollständigkeit, Eindeutigkeit und Widerspruchsfreiheit unmittelbar überwacht werden (Lánský, Ref. GPI-Kongreß Köln 1971).

Im Bereich des R U U gibt es mittlerweile auf einer Wortalgebra basierende algorithmische Verfahren zur Erkennung und Beurteilung schriftsprachlich frei formulierter Adressatenantworten - eine sehr bedeutende Lehrerfunktion -; das ermöglicht eine Überwindung des Antwort-Auswahlverfahrens bei der "klassischen" programmierten Unterweisung (Stahl 1970). Kybernetische Modelle des Gruppenlernens ermöglichen algorithmisch gesteuerte Unterweisung nicht mehr nur für Individuen, sondern auch für Kleingruppen bezüglich kognitiver und sozioemotionaler Aspekte, eine Funktion, die bislang nur vom personalen Lehrer vollziehbar erschien (Lánský 1971, Scharmann 1971).

Die folgende Untersuchung ist in zwei Abschnitte gegliedert. Wir versuchen zunächst die Ausgangssituation der rechnerunterstützten Lehrprogrammierung formalisiert zu erfassen, dann geben wir eine systemtheoretische Darstellung des Prozesses der Lehrprogrammierung.

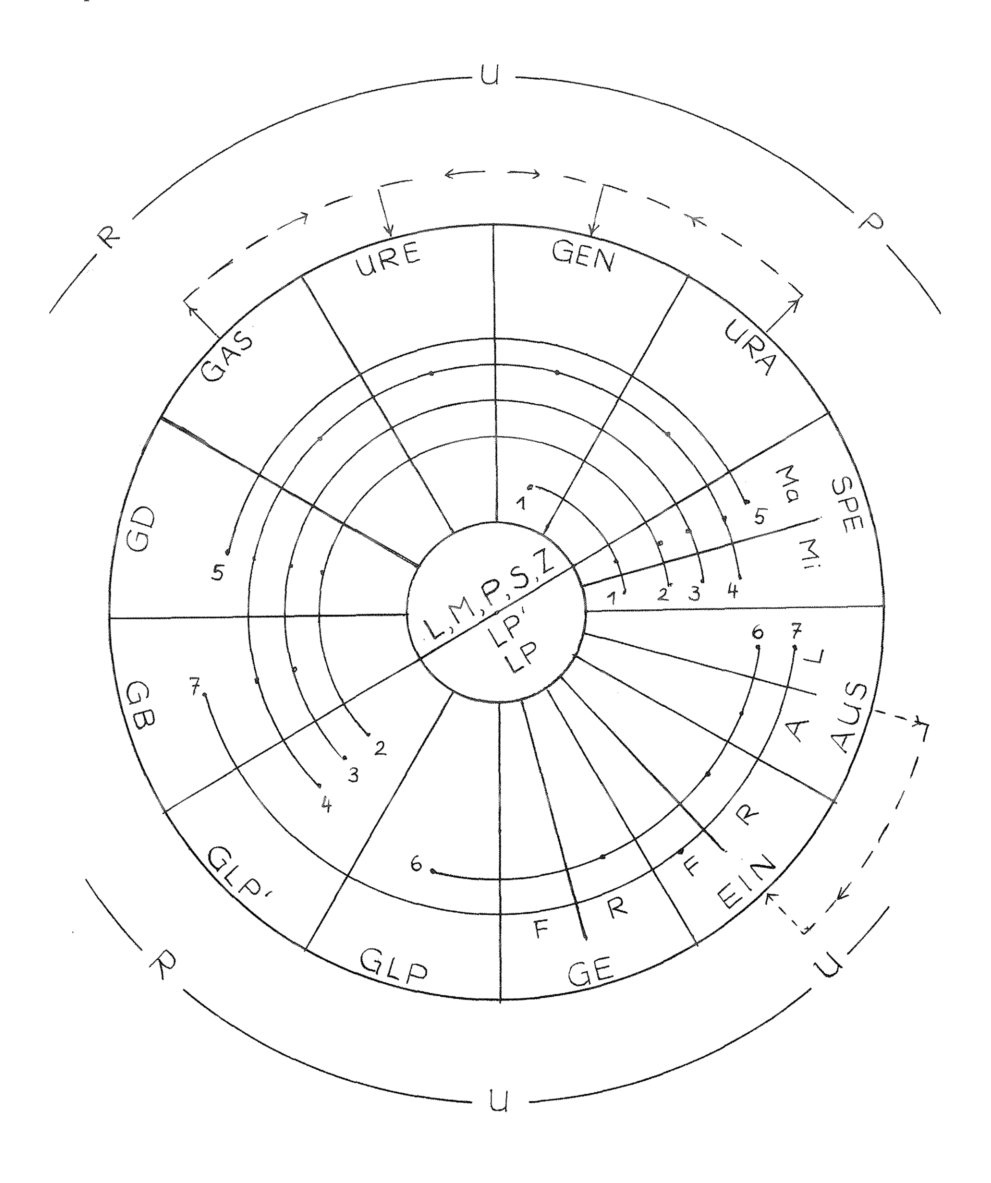


Abb. 1: Überblick über die objektivierten Lehrerfunktionen und die rechnerunterstützten Verfahren der Lehrprogrammierung und Unterweisung

Abschnitt A:
Die Ausgangssituation der rechnerunterstützten Lehrprogrammierung

I. Ansatz zu einer zusammenfassenden Darstellung der bisher bekannten RUP- und RUU-Verfahren. Zur Tragweite der personalen und der vollständig objektivierten Verfahren.

Wir haben in Abbildung 1 die im vorhergehenden angesprochenen z.T. algorithmierten Lehrerfunktionen graphisch zusammengefaßt und dabei die wichtigsten uns bekannten Verfahren der R U P und R U U markiert, die durch Verknüpfung entsprechender Bausteine entstanden sind. Dabei sind die Funktionen jeweils bezogen auf die Erzeugung bzw. Auswahl eines Lehrschritts im Zuge der Erstellung bzw. Darbietung eines vollständigen Lehrprogramms. Die beiden Bereiche R U P und R U U sind vereinigt, da mit zunehmender Objektivierung der Lehrerfunktionen bei diesen Aufgaben gewiß Lehrprogrammerzeugung und -darbietung zu einem Prozeß verschmelzen werden. Dies wird spätestens dann möglich, wenn didaktische Programme in der Art von Formaldidaktiken vorliegen, die im Verlaufe eines konkreten Unterweisungsprozesses nur aus den vorgegebenen didaktischen Daten (etwa in Form eines didaktischen Punktes (L, M, P, S, Z) im Sinne von Frank 1969) und den aktuellen Adressatenreaktionen das Lehrprogramm erzeugen, also ohne vorab programmierten Lehralgorithmus. Frank spricht hierbei von "Eigendidaktik". Die einzelnen Funktionen in der Abbildung 1 stützen sich jeweils auf die erwähnten Vorgegebenheiten, also die didaktischen Daten, dazu auf Bausteine (B) wie Lehrquanten, Verknüpfer, Grundformen, auf das bisher erzeugte bzw. abgewickelte Lehrprogramm (LP*), auf bisher erfolgte Reaktionen (R*) oder auf ein vollständig vorab programmiertes Lehrprogramm (LP).

Im einzelnen sind folgende Lehrerfunktionen erfaßt:

G — Generieren des nächsten Lehrschritts oder eines Bestandteils eines solchen, und zwar in zwei Stufen:
1. im Sinne der Makrostrukturgestaltung, d.h. im wesentlichen Festlegung des Lehrinhalts ohne Ausformulierung, und Einfügen in den Lehralgorithmus,
2. im Sinne der Mikrostrukturgestaltung, also fertige Ausformulierung.

Bei den einzelnen Verfahren, die in der Abb. 1 erfaßt sind, kann zuweilen nur eine dieser Stufen angesprochen sein.

GD — G unter Zugrundelegung des anfänglich gegebenen didaktischen Punktes D = (L, M, P, S, Z) im Sinne der formalen Didaktik nach Frank 1967.

GB	G unter Benutzung von Bausteinen zur Mikrostrukturgestaltung (vorgeg. Lehrquanten, Verknüpfer, Urteile).
GE	G unter Berücksichtigung von Eingaben des Adressaten im Verlaufe von R U U, dabei werden als Eingaben Reaktionen (R) auf Ausgaben hin (Aufforderung zur Reaktion) und frei gestellte Fragen (F) unterschieden.
GLP	G unter Berücksichtigung des vorhergehenden Verlaufes der R U U.
GLP	G unter Heranziehung der vorher festgelegten Lehrschritte eines vollständig vorbereiteten Lehrprogramms (d.h. also nur Auswahl eines Lehrschritts).
GAS	Ausgabe eines Lehrschritts oder Lehrschrittbestandteils, der durch G entstand, durch den Rechner in Mikro- oder Makrostrukturform zur Beurteilung durch einen Autor.
GEN	Entgegennahme eines Lehrschritts oder Bestandteils vom Lehrprogrammierer (Autor).
URA	Erzeugung und Ausgabe einer Reaktion (Urteil) durch den Rechner auf eine Eingabe (GEN) hin, z.B. auf Grund programmierter didaktischer Kriterien, für den Autor.
URE	Entgegennahme einer Reaktion (Urteil) des Autors auf eine Ausgabe (GAS) hin.
SPE	Einfügen eines kompletten Lehrschritts in den Lehralgorithmus bzw. ins Lehrprogramm (Speichern). Dies kann bzgl. der Makrostruktur allein erfolgen (Ma), oder unter Einschluß der Mikrostruktur (Mi).
AUS	Ausgabe eines Lehrschritts oder eines Bestandteils davon an einen Adressaten beim R U U, unterschieden nach L (reiner Lehrstoff) und A (Frage und Aufruf zur Reaktion).
EIN	Entgegennahme und Erkennen einer Eingabe eines Adressaten, entweder R (Reaktion) oder F (freie Frage).

Die bisher entwickelten und z.T. erprobten Verfahren der R U P und der R U U sind durch folgende Kombinationen charakterisiert:

1. RUP mit einer Autorsprache: GEN/SPE (Müller u. Wolber 1970)
2. RUP nach der Formaldidaktik ALSKINDI: GLP'/GD/SPE (Frank 1969 a)
3. RUP nach ALZUDI oder COGENDI: GLP'/GB/GD/SPE (Frank u. Graf 1967, Graf 1969, Blischke u.a. 1968)
4. RUP nach DIALOG-ALZUDI: GLP'/GB/GD/GAS/URE/GEN/URA/SPE: (Graf 1969 a, 1970)
5. RUP nach VERBAL: GD/SPE (Ma) (Lánský 1970, 1971 a)
6. RUU in Form von CAL: GLP/GE(R)/EIN(R)/AUS (Hartley u.a. 1970)
7. RUU als Auskunftssystem: GB/EIN(F)/AUS(L).

Die Tragweite der Objektivierungen bestimmter Lehrerfunktionen, insbesondere im R U P-Bereich, ist umstritten. Abgesehen von grundsätzlichen Erwägungen bildungstheoretischer Art, wie sie etwa von Nicklis 1967 vorgetragen wurden, oder didaktischer Art, wie Schulz 1967 sie anstellte, bestehen auch rein sachbezogen verschiedene Auffassungen darüber, in welchem Maße Rechner in den Bereichen von Darbietung und Erstellung von Lehrprogrammen effektiv eingesetzt werden können. Z.B. beweist das Arbeiten mit Autorsprachen und mit Formaldidaktiken, daß man einerseits glaubt, sich zunächst mit Protokollierungsleistungen von Rechnern begnügen zu müssen, während man andererseits bereits die algorithmische Herstellung von Lehrprogrammen vornehmen läßt. Dabei erscheint die erste Methode unnötigerweise auf zahlreiche algorithmische Kontrollmöglichkeiten zumindest methodischer Art zu verzichten, während sich die zweite mit sehr strengen Beschränkungen hinsichtlich der zulässigen Lehrstoffe und -ziele belastet.

II. Lehrprogrammieren im Autor-Rechner-Dialog

Ein konsequenter Ausweg aus der im Abschluß des letzten Kapitels angedeuteten Schwierigkeit scheint das Vorgehen nach einer "Dialog-Didaktik" zu sein. Diese berücksichtigt, daß sich für den Lehrprogrammautor in kurzen Abständen algorithmisch lösbare Aufgaben mit solchen abwechseln, die (zumindest gegenwärtig) nur personal zu bearbeiten sind. Eine Problemlösung unter solchen Voraussetzungen vollzieht man am besten in Form eines Dialogs zwischen den für die Lösung der jeweiligen Aufgaben zuständigen Systemen. Die erwähnte Integration unerläßlicher personaler Vollzüge in algorithmische Abläufe (oder umgekehrt), die das Wesen der elektronischen Datenverarbeitung bestimmen, ist das Hauptproblem bei allen heute gebräuchlichen rechnerunterstützten Programmierverfahren. Sie wird bei Autorsprachen und Formaldidaktiken meist durch getrennte Arbeitsgänge gelöst, die dann allerdings zu einer unnatürlichen Auflösung des Zusammenhangs der einzelnen Lehrerfunktionen führt. Etwa, wenn bei COGENDI zunächst personal Bausteine im wesentlichen ohne Bezug auf den Lehralgorithmus erstellt werden, um dann die Makrostruktur dieses Lehralgorithmus dazu algorithmisch erzeugen zu lassen.

Daß ein Autor-Rechner-Dialog hier angebracht ist, und in welchen verschiedenen "Stufen des Engagements" beider Systeme er vollziehbar ist, wurde in Graf 1970 bereits ausgeführt. Es seien hier nur die wesentlichsten Gedanken nochmals zusammengefaßt. Wir orientieren uns dabei hinsichtlich des Begriffs Dialog zwischen einem Menschen und einem Rechner an der Auffassung von Nees 1966, der diese Situation als "programmierten Dialog" so definiert: "Unter einem programmierten Dialog versteht man den gesprächsartigen Informationsaustausch zwischen

einem Menschen und einer Rechenmaschine, allgemeiner gesprochen einem Automaten. Der Informationsaustausch ist gesprächsartig, d. h. er bewegt sich in den gewohnten zeitlichen Dimensionen des normalen zwischenmenschlichen Gesprächs".

Es erscheint notwendig, vom programmierten Dialog noch zu verlangen, daß er auf bestimmte Gegenstände und auf ein bestimmtes Ziel bezogen ist. Nees läßt zwar diesen Aspekt außer acht und bringt als Beispiel einen programmierten Dialog, der sich auf kein bestimmtes Ziel zu richten scheint. Es wird jedoch spätestens bei den Formalisierungsansätzen, die in Teil B dieser Arbeit erfolgen, deutlich, daß programmierter Dialog nicht ohne Berücksichtigung der Sachzwänge beschrieben werden kann, die sich aus dem Dialog-Gegenstand und dem Dialog-Ziel ergeben. Diese Erkenntnis findet ihren Niederschlag in der Einführung eines Moderators für den Dialog.

Wir halten nochmals fest, daß - wie vorab im Überblick der Einleitung verdeutlicht wurde - die Menge der vom Rechner objektivierbaren Lehrerfunktionen reichhaltig genug ist, um diesen Informationsaustausch interessant zu machen, also über "Tischrechnerfunktionen" hinausreicht. Es sei dazu noch ergänzt, daß neben den in der Abb. 1 zusammengefaßten Verfahren, die ja sämtlich einen gewissen Vorrat methodischer oder didaktischer Rechneralgorithmen beinhalten, auch Strategien der Lehrprogrammierung, die nicht auf Rechnerunterstützung bezogen sind, häufig hochformalisierte Bestandteile enthalten. Das gilt z. B. für das Arbeiten mit Begriffsfortschrittsdiagrammen, Streckenkomplexen oder Stichwortschaubildern, die einen Einblick in Mikro- und Makrostruktur von Lehrprogrammen geben (Frank 1969, Lahn 1970, Zielke 1971). Viele didaktische Regeln schlagen sich in Anforderungen an die Struktur solcher Konstrukte nieder. Deren Überwachung kann, wickelt man die Protokollierung des zu erstellenden Lehrprogramms an einem Rechner ab, ohne großen Aufwand gleich von Rechnerprogrammen bewältigt werden. Auch hieraus wird also deutlich, daß ein Rechner ein interessanter Gesprächspartner beim Lehrprogrammieren werden kann.

III. Aufgaben der Theorienbildung. Anliegen der Untersuchung

Angesichts der in die Didaktik gehörenden zentralen Gegenstände der bisherigen Betrachtungen, nämlich Lehrprogrammen, ist es verständlich, daß auch beim Heranziehen von Rechnern zunächst noch die unterrichtstheoretischen Probleme im Vordergrund von Diskussionen und Publikationen standen. D. h. Probleme der Rechnerprogrammstruktur, der Programmiersprache u. ä. wurden in den Hintergrund gerückt bzw. ad hoc gelöst, indem z. B. die Formaldidaktiken nach Pro-

grammablaufplänen (Flußdiagrammen) in Assemblersprache programmiert wurden, deren logische Richtigkeit dann mit hohem Zeitaufwand am Rechner überprüft wurde. Diese pragmatische Vorgehensweise war zweifellos die vernünftigste, um möglichst rasch zu vorläufigen Ergebnissen über die Brauchbarkeit der R U P- und R U U- Verfahren zu kommen.

Nachdem sich nun abzeichnet, daß die rechnerunterstützten Verfahren der Lehrprogrammierung sich innerhalb bestimmter Grenzen bewähren und die Ergebnisse durchaus nutzbar sind, sollte man damit beginnen, diesen auch in formaler Hinsicht, d. h. also bezüglich der Theorie der Programmierung und der Programmiersprachen nachzugehen (Becker-Frank 1971, Blankertz 1969, Braun 1971, Closhen 1969, Graf 1967, Graf u. Illner 1970, Heinrich u. Weltner 1970, Waldau 1969, Weltner 1968).

Im einzelnen zeichnen sich in diesem Zusammenhang drei Problemkreise ab:

1. Didaktische Rechnerprogramme sind hinsichtlich ihrer logischen Struktur zu analysieren, Teilproblemblöcke sind möglichst differenziert zu isolieren. (Die Lösung dieses Problems zielt auf die Entwicklung von Rechnerprogramm-Bausteinen ab, die einerseits nach vershhiedenen Ansprüchen leicht kombiniert werden können, andererseits je nach den verschiedenen didaktischen oder methodischen Auffassungen leicht ausgetauscht werden können, ohne die Gesamtstruktur berühren zu müssen .)
2. Didaktische Rechnerprogramme sind daraufhin zu untersuchen, ob sie in jedem Falle eine Lösungsfindung gestatten. D.h. es ist zu prüfen, ob sie zu jedem formal richtig codierten Punkt ihres Definitionsbereiches (formal richtiger Input) eine Lösung in Form eines Lehrprogramms erzeugen können. Dies ist durchaus kein triviales Problem, auf die entsprechenden "Irrgartenprobleme" wies schon Frank 1966 hin.
3. Es ist eine an didaktischen Problemen orientierte Programmiersprache zu entwickeln, die die Formulierung didaktischer Programme (für R U P) und die Programmierung von Rechnern als Lehrautomaten (für R U U) gestattet. Weiterhin ist eine Dialogsprache zu entwickeln, die im Falle von Dialogverfahren sowohl bei der R U P als auch bei der R U U einen programmierten Dialog erlaubt. Einige nähere Ausführungen zu diesem Punkt finden sich in Graf 1970a.

Im Sinne dieser Problemstellungen ist das zentrale Anliegen der vorliegenden Arbeit die Erfassung und Beschreibung der formalen Struktur eines programmierten Dialoges mit dem Ziele der Erstellung von Lehrprogrammen. Wir benutzen dazu

ein geschlossenes System von endlichen determinierten abstrakten Automaten, darunter einen, der eine Formaldidaktik beschreibt.

Wir versuchen dabei zunächst, die Struktur des Dialogs möglichst allgemein z erfassen, d. h. die einzelnen Automaten ohne Bezug auf ein realisiertes Rechne programm zu beschreiben. Um die Brauchbarkeit dieser Struktur zu dokumen tieren, zeigen wir schließlich, wie die Automaten im Falle des bereits realisie ten Rechnerprogramms DIALOG-ALZUDI aussehen.

An Hilfsmitteln aus der Theorie der abstrakten Automaten benutzen wir nur ele mentare Begriffe und Definitionen, wie sie etwa in Frank 1969, Band I zusan mengestellt sind.

Abschnitt B:
Der Prozeß der Lehrprogrammierung in systemtheoretischer Darstellung.

I. Lehrprogrammieren als Regelungsvorgang: Belehrung eines Psychostrukturmodells.

Wir legen den Ansatz von Frank zugrunde, wonach die uns interessierende Didaktik durch einen Definitionsbereich $\mathfrak{D}_{\{L, Z, M, P, S\}}$ im durch die Dimensionen Lehrstoff, Lehrziel, Lehrmedium, Psychostruktur und Soziostruktur bestimmten didaktischen Raum, durch eine Menge $\{\Lambda\}$ von Lehralgorithmen und durch eine Abbildung

$$D : \mathfrak{D}_{\{L, Z, M, P, S\}} \rightarrow 2^{\{\Lambda\}}$$

bestimmt ist.

D.h. insbesondere, L und Z sind vorgegeben und damit nicht mehr Probleme der Lehrprogrammierung (Frank 1969, I. S. 360)

Lehrprohrammieren bedeutet in diesem Zusammenhang die Ausführung dieser Abbildung bei vorgegebenen Werten der fünf Variablen L, Z, M, P, S und das Anschließen einer weiteren Abbildung von der Lösungsmenge von Lehralgorithmen auf ein Lehrprogramm, also eine Ausformulierung des Lehralgorithmus in einer konkreten Sprache, oder in der Frankschen Notation eine zu einem Lehralgorithmus $\Lambda = \{\mathfrak{R}, \mathfrak{Y}, \varphi\}$ aus $D(\mathfrak{D})$ gehörige Menge komplexer Lehrschritte.

Die sich uns stellende Aufgabe ist die Ausführung dieser Abbildungen. Dabei bewährt sich das Prinzip der Regelung in folgender Weise: Man ersetzt zunächst den Adressaten durch seine Psychostruktur P, beschrieben durch ein (automatentheoretisches) Psychostrukturmodell (PYM), und betrachtet dieses als Lernsystem. Der Lehrprogrammautor oder dessen Objektivation, die uns hier besonders interessiert, wird als Lehrsystem betrachtet und ebenfalls durch einen abstrakten Automaten dargestellt. Die kreisrelationale Verknüpfung von Lehr- und Lernsystem zu einem geschlossenen dynamischen System von zwei abstrakten Automaten kann nun als Regelkreis gesehen werden, in dem das Lehrsystem das Vergleichsglied und den Regler, das Lernsystem das Regelobjekt darstellen. Zum Anfangszustand des Systems gehören der Anfangszustand des PYM sowie die Daten eines Punktes aus $\mathfrak{D}$.

In diesem Regelkreis, den wir in Abbildung 2 dargestellt haben, läßt sich nun der von Frank 1966 vorgeschlagenen Weise das gesuchte Lehrprogramm (als E gebnis der beiden genannten Abbildungen) konstruieren, indem pro Regeltal ein komplexer Lehrschritt vom Regler erzeugt wird und auf seine Brauchbarke für die Gesamtlösung hin am Lernsystem, hier also am Modell der Psychostruk getestet wird.

Das "Testen" besteht jeweils aus der Vorausberechnung von Bestandteilen d Lehralgorithmus bzw. des Lehrprogramms (Lehrquanten, Fragen usw.) durch d Regler, die geeignet sind, das Modell in Richtung auf den Zielzustand zu ve ändern. Die Berechnungen stützen sich auf Kenntnisse über interne Reaktion (d. h. hier Zustandsänderungen) des PYM auf solche Bestandteile und auf die A ihrer Abfolge hin, also unter dem Einfluß der Mikro- und der Makrostruktur d Lehrprogramms. Man kann diese Vorgehensweise auch so interpretieren, daß i Rahmen einer Regelung das PYM selbst belehrt wird, bis es den vorgegebene durch Z bestimmten Zielzustand erreicht. Das gesuchte Lehrprogramm beste dann in einem Protokoll dieser Belehrung.

Der Unterschied zur bei Unterrichtsmodellen üblichen Verkoppelung von Leh und Lernsystem besteht darin, daß sich hier beim Lehrprogrammieren das Leh system nicht auf externe Reaktionen des Lernsystems stützen kann (Antwort o. ä., also auf die Ergebnisfunktion des Lernsystems), sondern nur auf die The rie über die internen Reaktionen (Überführungsfunktion).

Bisher ist noch völlig offen gelassen, in welcher Weise die beiden genannten a strakten Automaten realisiert werden. Nur die grundsätzliche Strategie der L sungsfindung wurde festgelegt. Die Bestandteile des Regelkreismodells könn dagegen mit beliebigen Systemen besetzt werden, soweit diese als abstrakte A tomaten darstellbar sind. Wenn diese Voraussetzung erfüllt ist, sind also beli bige Psychostrukturmodelle für das Lernsystem zulässig. Ebenso kann die Fun tion des Reglers sowohl von einem personalen Autor als auch von einem Rechn mit geeignetem Steuerprogramm oder auch von einer geeigneten Kombinati beider übernommen werden. Realisiert sind die genannten Möglichkeiten ta sächlich durch Autorsprachen, Formaldidaktiken und die Dialog-Didaktik.

Dem Anliegen dieser Arbeit entspricht es nun speziell, eine Verwirklichung d letzten Falles zu untersuchen, den (Autor-Rechner-)Dialog-Regler. Dieser en hält übrigens einen vollprogrammierten Regler, nämlich die Formaldidaktik A ZUDI 2 als Spezialfall.

Für das Lernsystem verwenden wir ein an Frank 1966 angelehntes Psychostrukt modell unter Einbeziehung einiger lerntheoretisch-methodischer didaktisc Aspekte.

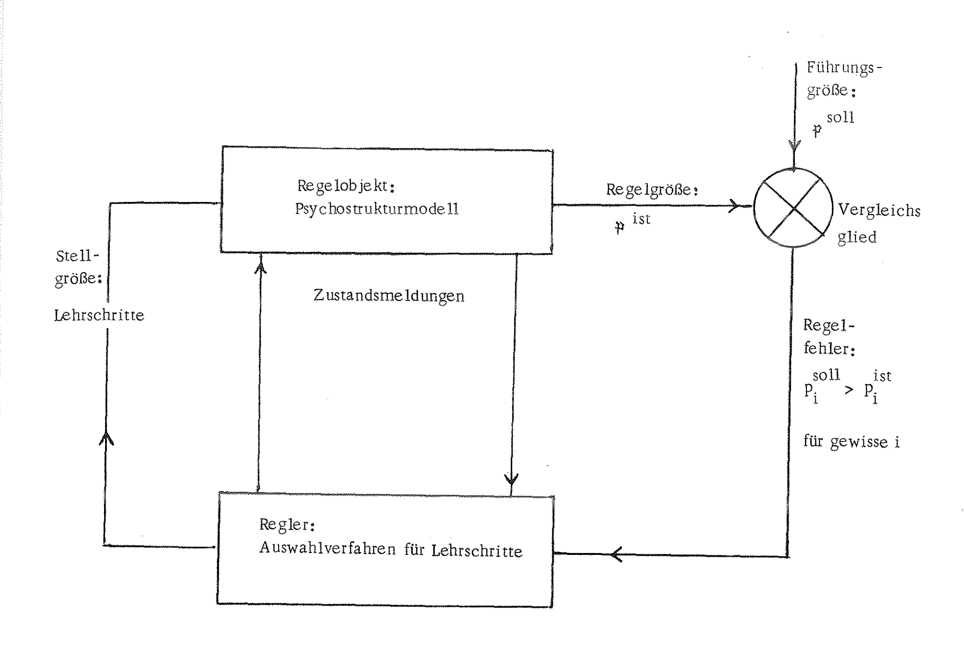


Abb. 2: Lehrprogrammieren als Regelung

II. Lehrprogrammieren im programmierten Autor-Rechner-Dialog

In einem ersten Schritt der Spezialisierung unseres Ansatzes aus Kapitel I des Abschnitts B wollen wir nun im Sinne der in der Einleitung genannten Vorzüge des programmierten Dialogs das Regelkreismodell konkretisieren, indem wir von einer unterrichtstheoretischen Situation ausgehend, d. h. von einem Definitionsbereich einer Didaktik, das Lernsystem durch ein konkretes Psychostrukturmodell PYM und den Regler durch ein Autor-Rechner-System beschreiben.

II. 1 Formalisierte didaktische Ausgangssituation; Daten der vorgegebenen Punkte (L, Z, M, P, S) des Definitionsbereiches.

a) Lehrstoff und ästhetische Information: Der Lehrstoff L besteht aus einer Menge L von g Zuordnungen zwischen beliebigen Begriffen, codiert durch die ganzen Zahlen 1,2,..., g. Die Zahlengröße bestimmt gleichzeitig die Reihenfolge, in der die Zuordnungen dem PYM durch das Lehrprogramm bekanntzumachen sind. Für nicht mehr als die Hälfte der Zuordnungen kann noch vorgeschrieben sein, daß sie dem PYM nur in Form von Fragen vorzulegen sind. Für einige oder alle Begriffe bzw. Zuordnungen kann verlangt sein, daß sie nach der ersten Erwähnung in einem Lehrquant sofort in Frageform wiederholt werden.

Nicht als eigentlicher Lehrstoff, sondern als Bausteine für die Herstellung des Lehrprogramms sind weiterhin vorgegeben eine Menge G von "Basaltextsätzen", die die Zuordnungen aus L vollständig erklären, eine Menge F von "Grundformen", das sind Sätze, die die Zuordnungen aus L kurz beschreiben bzw. danach fragen und eine Menge S von "Schlußbemerkungen", die die Behandlung bestimmter Zuordnungen im Lehrprogramm abschließen sollen.

b) Lehrziel: Für jede Zuordnung i aus L ist eine Mindestwahrscheinlichkeit p_i^{soll} vorgegeben, mit der i nach Ablauf der Belehrung gelernt sein soll.

c) Psychostrukturmodell: Als Anfangszustand ist für jede Zuordnung i aus L eine Anfangswahrscheinlichkeit p_i^0 vorgegeben, mit der i im PYM bereits bekannt ist. PYM kann in einem Zeittakt jeweils eine Zuordnung aus L aufnehmen, verbunden mit einer geeigneten ästhetischen Form aus G, F oder S. (zur Bedeutung der ästhetischen Information, hier getragen von "ästhetischen Formen" vgl. Frank u. Frank-Böhringer 1967). Dabei ist mit jedem Takt zu wechseln zwischen der Form eines Lehrquants und der einer Frage, zu beginnen ist mit einer Lehrquantenform.

Wird die Zuordnung i zum n-ten Male an PYM gegeben, so berechnet sich unter der Voraussetzung, daß die untenstehenden Bedingungen $c_1 - c_4$ erfüllt sind, die Wahrscheinlichkeit p_i^{ist}, mit der i nunmehr bekannt ist, nach einer Funktion, die abhängt von der Zuordnung i selbst, von der damit verbundenen ästhetischen Form ä, von p_i^{ist} (n-1) und von gewissen zeitlich konstanten informationspsychologischen Größen, die sich auf die Kapazitäten von PYM beziehen, also $p_i^{ist}(n) = f(i, ä, p_i^{ist}(n-1), k_1, k_2, \ldots)$. Die Funktion liefert für n = 0 den Wert p_i^0, mit wachsendem n konvergiert die Folge der Werte von f monoton steigend gegen 1. Beispiele für die Gestalt von f sind:

1. $p_i^{ist}(n) = 0{,}87 \cdot p_i^{ist}(n-1) + 0{,}13$ (ALZUDI 1, Frank u. Graf 1967),
2. $p_i^{ist}(n) = 1 - (a/b)^{cb/d}(1 - p_i^{ist}(n-1))$ mit $a = 1{,}5 \cdot B$, wenn B die Länge der zugehörigen ästhetischen Form ist, $b = i_i(n-1) + 1{,}5 \cdot B$, wenn $i_i(n-1)$ die Information von i nach der (n-1)-ten Erwähnung von i ist, $c = c_{vk}/c_k$, der Quotient der Zuflußgeschwindigkeiten in vorbewußtes Gedächtnis und Bewußtsein, $d = i_i(n-1)$ (ALZUDI 2, vgl. Graf 1969).

c_1: Die Zuordnung i kommt in den drei vorhergehenden Takten nicht als Eingabe in PYM vor.

c_2: Nach der neuen Eingabe muß gelten: $0 \leq q-f \leq 2$, wenn die Zuordnung i insgesamt q mal in Lehrquantenform und f mal in Frageform an PYM gegeben wurde. (Dies gilt natürlich nicht für Zuordnungen, die nur als Fragen auftreten sollen.)

c_3: Ist h die Anzahl der bereits eingeführten Zuordnungen, so muß für i gelten: $p_i^{ist}/p_i^{soll} \leq C(h, g)$ oder: i kommt erstmals als Eingabe in PYM vor. Dabei ist C eine Funktion von h und g mit dem Wert 1 sobald h = g. Beispiel für die Funktion C: $C(h, g) = (h/g)^{1/\alpha}$ mit beliebiger ganzer Zahl α (ALZUDI 2 Graf 1969).

c_4: Die Zuordnung i soll nicht in einer zeitlichen Eingabefolge von Zuordnungen stehen, die im Verlaufe des Lehrprogramms schon einmal aufgetreten ist.

Die Bedingungen $c_1 - c_4$ lassen sich auch mathematisch-lerntheoretisch fassen, sie wurden jedoch absichtlich in dieser Form aufgenommen, um zu zeigen, daß auch nichtnumerische methodische Regeln Bestandteil von PYM sein können.

Es ist noch zu bemerken, daß der Lernzuwachs, also die Zunahme von p^{ist}, sowohl erfolgt, wenn eine Zuordnung in einem Lehrquant an PYM gegeben wird, als auch wenn diese in Form einer Frage vorkommt. Im letzteren Falle wird der erste Begriff als Reiz gegeben, begleitet von einer ästhetischen Frageform ä, der zweite ist zu finden. Wir gehen also davon aus, daß ebenso beim Lehren wie auch beim Beantworten von Fragen gelernt wird. Aus diesem Grund erfolgt die Eingabe in PYM auch nicht lehrschrittweise, wie im ursprünglichen Frankschen Modell, sondern im Wechsel von Lehrquanten und Fragen, letztere bereits verbunden mit dem bestätigenden richtigen Antwortbegriff.

d) Die Größen Medium und Soziostruktur werden nicht explizit festgelegt.

e) Lehralgorithmus und Lehrprogramm: Es ist ein Lehrprogramm aufzufinden, - hier in Form einer Kette von mit ästhetischen Formen verbundenen Zuordnungen -, unter dessen Einfluß PYM das Lehrziel Z erreicht, also in einen Zielzustand übergeht.

Die Begründung und nähere unterrichtstheoretische Erläuterung für den im vorstehenden beschriebenen Definitionsbereich unseres Lehrprogrammverfahrens soll hier unterbleiben, da es uns nur um die formale Erfassung geht. Sie sind in Graf 1969 gegeben.

II. 2 Der Regler als Lehrsystem und Vergleichsglied

Nachdem die theoretische Ausgangssituation für den Prozeß der Lehrprogrammierung erfaßt ist, und damit auch die Gestalt des Lernsystems, hier also des Psychostrukturmodells, ist nunmehr die Struktur des Reglers festzulegen, der die Rolle des Lehrsystems übernehmen kann. Der Regler hat die Aufgabe, in jedem Zeittakt eine Zuordnung i aus L zusammen mit einer ästhetischen Form bereitzustellen, so daß eine Eingabe für PYM entsteht, die dort zu einem Lernzuwachs führt, bzw. zu einer Zustandsänderung in Richtung auf den Zielzustand. Es ist zweckmäßig, davon auszugehen, daß dem Regler zu diesem Zwecke stets die vorgegegebenen Datenmengen G, F, S und L zur Verfügung stehen, ebenso in jedem Takt die aktuellen Zustandsdaten von PYM. Nun ist es zwar verhältnismäßig einfach, ein Lehrprogramm aufzustellen, das durch genügend häufiges Wiederholen der Zuordnungen jeden Wert p^{ist} von p^0 ausgehend über p^{soll} hinausführt. Tatsächlich aber entstehen Schwierigkeiten dadurch, daß aufgrund der in II. 1 c genannten Bedingungen $c_1 - c_4$ das Psychostrukturmodell in Zustände geraten kann, die jeweils gewisse Zuordnungen als zu keinem Lernzuwachs führend als Eingabe ausschließen. Die Berücksichtigung dieser Tatsache ist gewissermaßen die didaktische Leistung des Reglers.

Wir greifen, wie erwähnt, die in Abschnitt A erwähnten Argumente für den Dialog auf und wollen deshalb als Regler ein Autor-Rechner-System entwerfen, das die verlangten Regelleistungen erbringen kann.

Den Bedürfnissen des PYM entsprechend hat dieses System also in jedem Zeittakt eine Zuordnung aus L und eine zugehörige ästhetische Form aus G oder F in Verbindung mit S auszuwählen. Wir wollen dies in zwei aufeinanderfolgenden konstruktiven Arbeitsgängen vornehmen lassen. Wir gehen davon aus, daß grundsätzlich Autor und Rechner beide Auswahlen treffen können, jeweils unter Berücksichtigung der für sie geltenden Kriterien.

Weiterhin setzen wir voraus, daß beide Teilsysteme des Reglers Lösungsvorschläge des anderen für die Auswahl einer ästhetischen Form oder einer Zuordnung auf ihre Verträglichkeit mit den eigenen Kriterien überprüfen können. Auf diese Weise können die charakteristischen Fähigkeiten der Partner zur Bewältigung beider Auswahlprobleme durch Konstruktion und Prüfung zum Tragen kommen.

Speziell in unserem Falle - das dürfte aber auch für viele andere Dialog-Lösungsverfahren zutreffen - kann sich der Dialog nicht völlig ungebunden entfalten, sondern wird wesentlich vom Dialog-Ziel beeinflußt, d. h. hier von der Aufgabe, ein Lehrprogramm zu vorgegebenen Daten zu erstellen. Dieses Ziel schreibt eine Abfolge von Arbeitsschritten im Dialog vor, die ihrerseits formalisierbar ist. Es erscheint deshalb zweckmäßig, die Leitung des Dialogs nicht dem Autor zu überlassen, da dieser sonst eine Vielzahl sich wiederholender Aufrufe an den Rechner zu geben hätte, sondern diese Aufgabe ebenfalls zu objektivieren und an den Rechner zu übertragen. D. h. in unseren Regler wird als Bestandteil ein Moderator MOD aufgenommen, der im Interesse des Dialog-Ziels jeweils bestimmt, welche Leistungen von welchem Partner zu erbringen sind.

Damit wird das Blockschema der Abbildung 3 deutlich, das den Regelkreis der Abbildung 2 für den Fall eines Dialog-Reglers spezifiziert. Sich am Regelfehler orientierend, veranlaßt der Moderator dabei jeweils Auswahl und Prüfung einer geeigneten Zuordnung bzw. einer zu einer bereits ausgewählten Zuordnung passenden ästhetischen Form. Ablehnungen werden dem Moderator und der Quelle zur Kenntnis gebracht, bei Übereinstimmung für beide Teile erfolgt aus einem Zwischenspeicher für die Stellgröße heraus die Beeinflussung des PYM.

Dem gesamten Reglersystem stehen jeweils die aktuellen Zustandsdaten des PYM und die didaktischen Ausgangsdaten zur Verfügung.

Der Ablauf des Dialogs soll nun durch den programmierten Moderator nicht unveränderlich bestimmt werden. Zu diesem Zwecke wird dem Autor bei der Moderierung des Dialogs ein Mitspracherecht eingeräumt. Dessen Wahrnehmung erfolgt in der Weise, daß der Moderator mit dem Autor einen Meta-Dialog neben dem Objekt-Dialog über die Vorgehensweise führen kann, z.B. hinsichtlich der Frage, welcher Dialogpartner als nächster aktiv werden soll. Damit ist neben dem programmierten Moderator MOD noch ein Autor-Moderator AMO im Regler vorhanden.

III. Die Funktion des Dialog-Moderators MOD

Unterscheiden wir hinsichtlich der Aufgaben Auswahl und Prüfung, die jeder der Dialog-Partner vollziehen kann, jeweils noch nach der für eine Zuordnung und der für eine ästhetische Form, so umfaßt unser Dialog-Regler nunmehr insgesamt zehn Teilsysteme:

MOD	(Moderator), der programmierte Dialog-Moderator,
AMO	(Autor-Moderator), der Autor als gelegentlich hinzugezogener Zusatz-Moderator,
RWZ	(Rechner wählt Zuordnung), der Rechner als Auswählender einer Zuordnung,
RWA	(Rechner wählt ästhetische Form), der Rechner als Auswählender einer ästhetischen Form,
RPZ	(Rechner prüft Zuordnung), der Rechner als Prüfer einer Zuordnung,
RPA	(Rechner prüft ästhetische Form), der Rechner als Prüfer einer ästhetischen Form,
AWZ	(Autor wählt Zuordnung), der Autor als Auswählender einer Zuordnung,
AWA	(Autor wählt ästhetische Form), der Autor als Auswählender einer ästhetischen Form,
APZ	(Autor prüft Zuordnung), der Autor als Prüfer einer Zuordnung,
APA	(Autor prüft ästhetische Form), der Autor als Prüfer einer ästhetischen Form.

Daneben existieren Speicher L, G, F und S für die vorgegebenen Daten und ein Ausgabespeicher AUS mit den Teilen AUSr und AUSä für die jeweils zur Diskussion stehende Zuordnung bzw. ästhetische Form.

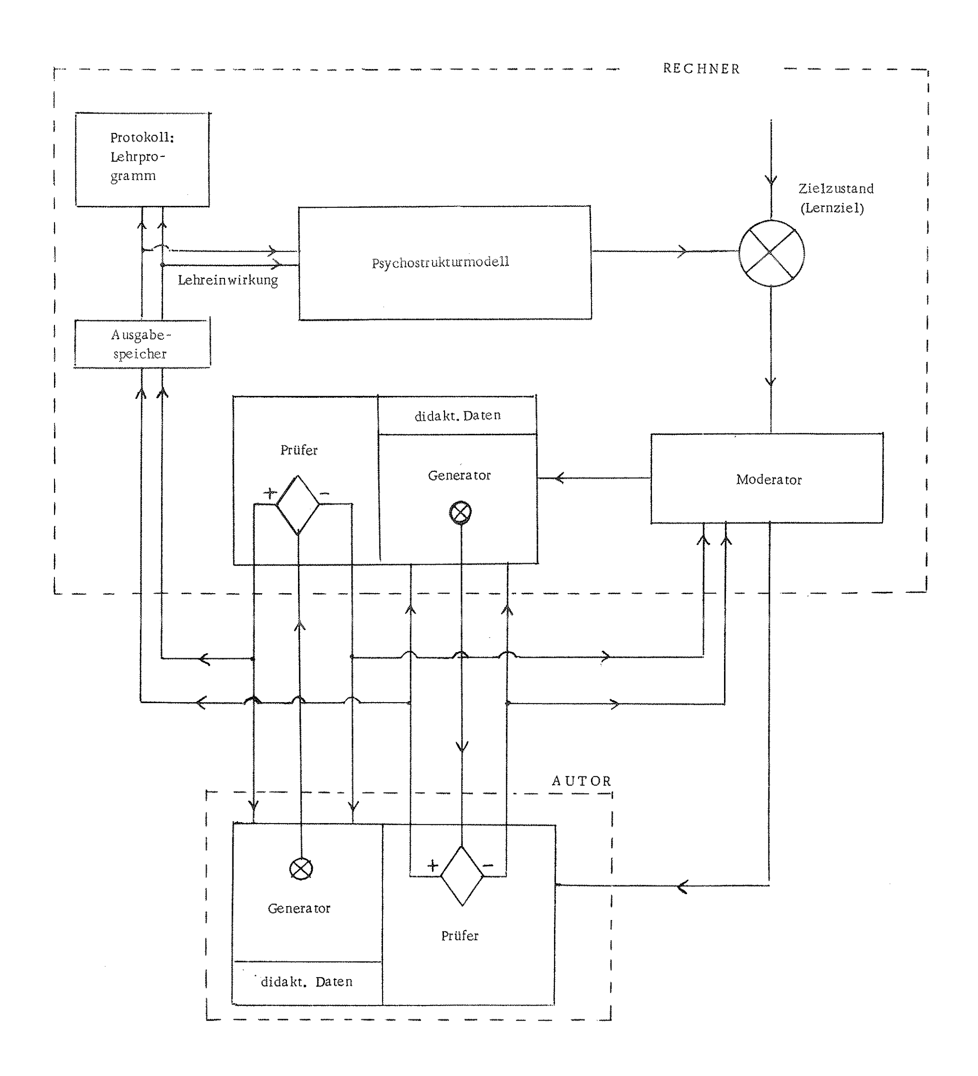


Abb. 3: Grundstruktur eines Autor-Rechner-Dialog-Systems zur Lehrprogrammierung

Die Konstruktion der Stellgröße, bestehend aus einer Zuordnung und einer ästhetischen Form, die auf PYM einwirken soll, erfolgt in jedem Takt prinzipiell nach dem Ablaufplan, den Abbildung 4 zeigt. Dabei gilt bezüglich der Taktung: Im Rahmen des Regelkreises PYM - Dialog-Regler erfolgt die Auswahl einer Stellgröße in einem Funktionstakt. Ein solcher beginnt mit der Eingabe der Stellgrößen in PYM. Die Funktion des Reglers selbst umfaßt mehrere Teiltakte (von Fall zu Fall verschiedene Anzahl, abhängig von der Art und Anzahl der "Äußerungen" der Partner), die jeweils mit einer Eingabe in den Moderator beginnen.

Die Punkte (1) bis (7) in Abb. 4 beziehen sich jeweils auf eine Aktivität des Moderators MOD. Die Kriterien für die Verzweigungen werden in Kapitel IV gegeben (Abb. 5).

Erläuterung des Ablaufs: Nach einem Anstoß von PYM veranlaßt der Moderator (1) die Auswahl einer Zuordnung durch den Rechner (RWZ) oder den Abbruch der Regelung, falls darüber Einigkeit besteht (STOP). Es folgt (2) die Prüfung der Zuordnung durch den Autor (APZ). Daran schließt sich eine Rücksprache von MOD (3) mit dem Autor als Zusatzmoderator an (AMO). Diese führt entweder erneut zu RWZ oder zu AWZ, oder MOD (5) ruft abhängig vom Ergebnis der Rücksprache zur Auswahl einer ästhetischen Form durch den Rechner oder durch den Autor auf, (RWA oder AWA).

Auf AWZ folgt (4) RPZ und dann Rücksprache wie oben (3), auf RWA folgt (6) APA, auf AWA (8) RPA. Bei Ablehnung einer vom Rechner selektieren ästhetischen Form stößt der Moderator auf jeden Fall gleich AWA an (7).

Besteht Einigkeit über Zuordnung und ästhetische Form, so kann der Moderator (7) die Ausgabe der Regelgröße, nämlich den Inhalt von AUS an PYM veranlassen und somit einen Funktionstakt des Reglers beenden.

Damit haben wir in einer ersten Stufe der Konkretisierung auch die Struktur und Funktion eines Reglers für unseren Lehrprogrammier-Regelkreis angegeben, der in jedem Zeittakt eine Regelgröße an das Regelobjekt PYM liefern kann und sich dabei auf eine optimale Zusammenarbeit zwischen Autor und Rechner stützt.

Natürlich ist in der Festlegung des Funktionsablaufs eine methodische Willkür enthalten. Davon kann jedoch bei anderen Lösungen abgewichen werden, ohne daß unsere Blockstruktur des Reglers verändert werden muß.

Im folgenden IV. Kapitel wollen wird die genannten zehn Teilsysteme ebenso wie das Psychostrukturmodell PYM nun tatsächlich als abstrakte Automaten angeben und damit die Struktur eines Rechnerprogramms bereitstellen, das tatsächlich die Abwicklung eines Regelvorganges "Lehrprogrammierung" im Autor-

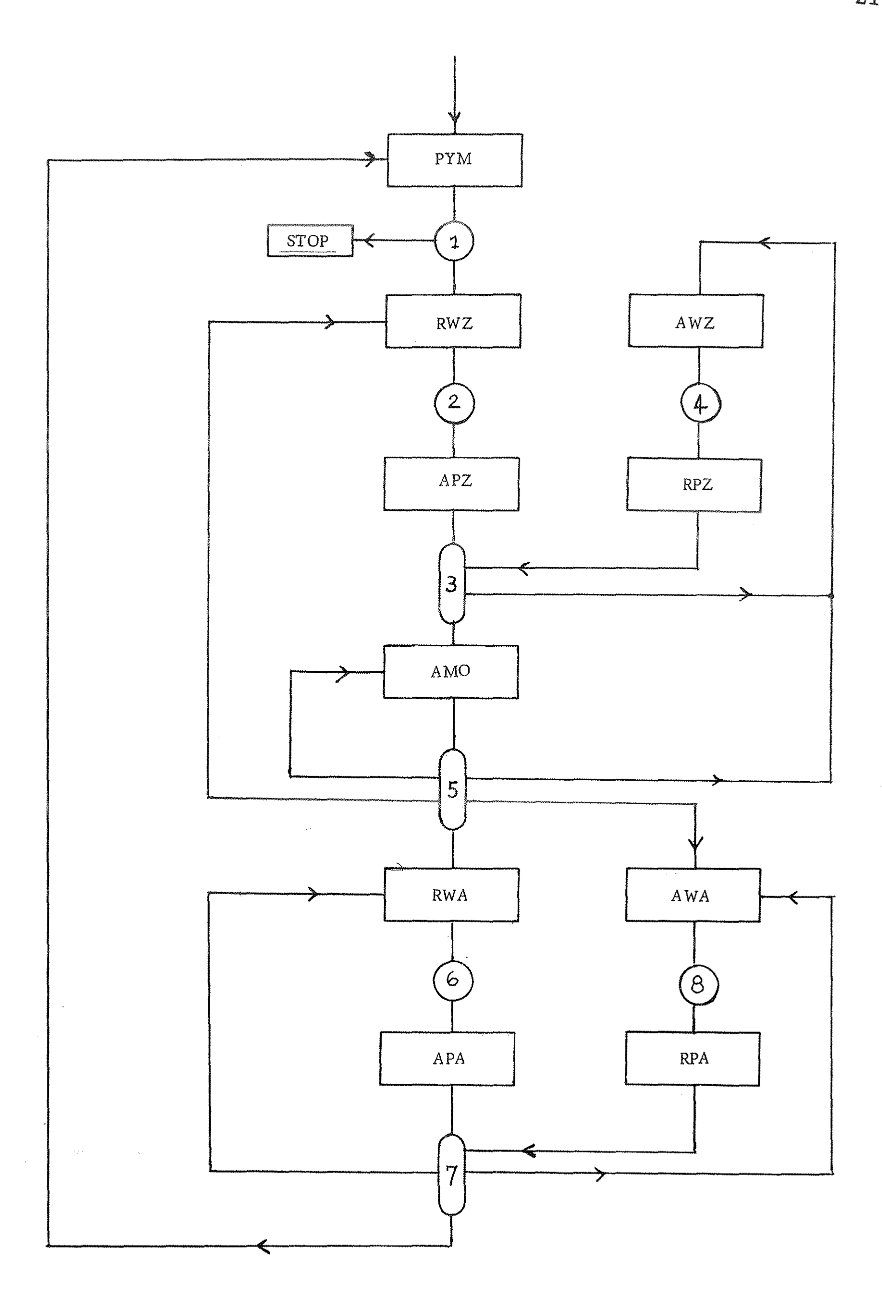


Abb. 4: Funktionsmodus des Moderators

Rechner-Dialog gestattet. Mit einigen geringfügigen Abweichungen liegt dieses Programm "DIALOG-ALZUDI" in der Assemblersprache PROSA codiert bereits vor (Graf 1970).

Damit führen wir eine zweite Stufe der Spezialisierung unseres ursprünglichen Ansatzes durch, indem wir jeweils eine bestimmte Funktionsweise der Teilsysteme auswählen. Selbstverständlich kann auch auf dieser Stufe wieder eine andere Auswahl getroffen werden, ohne daß das bislang dargestellte Modell des Dialogs verändert werden muß.

IV. Konkretisierung des Lehrprogrammier-Regelkreises durch das Rechnerprogrammsystem DIALOG-ALZUDI

IV.1 Formalisierte didaktische Ausgangsdaten (ohne PYM)

a) Lehrstoff und ästhetische Information: Es sind maximal zehn Mengen von Begriffen gegeben. Der Lehrstoff besteht in einer Menge L von Zuordnungen zwischen je zwei Begriffen aus diesen Mengen. Jede Zuordnung aus L wird beschrieben durch ein 13-tupel von (numerischen und nichtnumerischen) Daten.

1. Rangnummer r, eine natürliche Zahl, im folgenden stellvertretend für die Zuordnung verwendet, bestimmt die Reihenfolge der Einführung in das Lehrprogramm.
2. t', der erste Begriff der Zuordnung; $/t'/$ dessen Länge.
3. t'', der zweite Begriff der Zuordnung.
4. Typ t der Zuordnung r, eine zweistellige natürliche Zahl, bezieht sich auf die Mengen, aus denen t' und t'' stammen.
5. Basaltextsatz-Nummer b', der Index des Basaltextsatzes der Menge G (s. u.), in dem r erläutert wird, $b' = 0$ bedeutet, daß kein solcher Satz vorliegt.
6. Schlußsatz-Nummer s', der Index des Schlußsatzes aus der Menge S, der die Behandlung von r im Lehrprogramm abschließt; $s' = 0$ bedeutet, daß kein solcher Satz vorliegt.
7. m, eine Boolesche Größe, die angibt, ob r nach Einführung in einem Lehrquant sofort in der Frage zu wiederholen ist (m = 1), oder ob die Frage "leer" folgen soll, d. h. Zuordnungs-Nummer 0 und leere ästhetische Form.
8. n, eine Boolesche Größe, gibt an, ob eine Zuordnung in Lehrquanten und Fragen zu verwenden ist (n = 1), oder nur in Fragen.
9. q, eine Boolesche Größe, q = 0 bedeutet, daß $b' = 0$ (vgl. Nr. 5)
10. a', eine Boolesche Größe. $a' = 0$ bedeutet, daß der Basaltextsatz, der r erklärt, bei der Einführung von r ins Lehrprogramm nicht benutzt werden soll.
11. e', eine Boolesche Größe. $e' = 0$ bedeutet, daß der Basaltextsatz, der r erklärt, bei der letzten Erwähnung von r im Lehrprogramm nicht benutzt werden soll.

12. p^o , die Speicherwahrscheinlichkeit von r in PYM vor Beginn der Belehrung.
13. p^{soll}, der Mindestwert der Speicherwahrscheinlichkeit von r in PYM nach der Belehrung, "Sollwert".

Zur Vereinfachung der folgenden Darstellungen ergänzen wir die Menge L um eine "leere" Zuordnung mit der Rangnummer 0, die verwendet wird, wenn ein Lehrquant oder eine Frage ausgelassen werden soll oder eine ästhetische Form ohne direkten Bezug zu L enthalten soll. Wir definieren dazu die Daten des 13-tupels sämtlich gleich 0, mit Ausnahme von $n_0 = 1$ (vgl. Nr. 8).Wir setzen $L_0 = L \cup \{0\}$.

Die g+1 Daten jedes einzelnen Typs können wir als (g + 1)-dimensionale Vektoren betrachten, wir haben dann insbesondere $\mathfrak{m}$, $\mathfrak{n}$, $\mathfrak{p}^0$ und $\mathfrak{p}^{soll}$.

Wir setzen voraus: Umfang von $L = /L/ = g \geqq 4$, Anzahl der 1-Komponenten von $\mathfrak{n} = /\mathfrak{n}/ \geqq g/2$, $n_r + m_r = 1$ für alle r aus L.

Weiter liegt vor eine Menge G von Basaltextsätzen $\{g_{b'}\}$ und eine Menge S von Schlußbemerkungen $\{s_{s'}\}$.

Schließlich gibt es die Menge F von Grundformen, d.h. unvollständigen Sätzen, die durch Einsetzen der Begriffe von Zuordnungen des passenden Typs zu vollständigen Lehrquanten oder Fragen werden. Jede Grundform ist beschrieben durch ein Tripel von Daten:

1. h', die Grundform selbst,
2. Typ t der Grundform, wie Datum 4 bei den Zuordnungen, gibt an, Zuordnungen welchen Typs zur Vervollständigung der Grundform in Frage kommen.
3. x', mit $x' = 1$, falls aus der Grundform ein Lehrquant entsteht, $x' = 0$, falls eine Frage.

b) Das Lehrziel ist durch den Vektor $\mathfrak{p}^{soll}$ gegeben.

IV.2 Das Psychostrukturmodell PYM als initialer abstrakter Automat mit diskreten Zeittakten t (Regelobjekt)

a) Eingabebuchstaben x_{PYM} = (PYM!). (r, i, f, ä, A, B, D) sind die Stellgrößen des Reglers, also dessen Ausgabebuchstaben y_{REG}. Wir takten so, daß gilt: $x_{PYM}(t) = y_{REG}(t-1)$.

Dabei ist r aus L_0 oder $r = g'$, das Abschlußzeichen, $i = 1$ gibt an, daß r erstmals angeboten wird, sonst $i = 0$, $f = 1$ gibt an, daß r letztmalig erwähnt wird,

sonst $f = 0$, ä ist die ästhetische Form, in die r eingebettet wird, konstruiert aus G oder F und S; A, B und D sind Werte von Booleschen Zustandsgrößen des Reglers, bzw. von RWZ (vgl. IV.4).

b) Ausgabebuchstabe y_{PYM} = MOD!, der Aufruf an den Regler (faktisch den Moderator), den Arbeitstakt zu beginnen. (Man könnte auch die Wertemengen der Zustände von PYM als Ausgabebuchstaben auffassen, jedoch ist es zweckmäßiger, davon auszugehen, daß die aktuellen Zustände von PYM allen Teilsystemen des Reglers ohne expliziten Datentransport zugänglich sind .)

c) Zustände $z_{PYM} = (\mathfrak{p}^{ist}, \mathfrak{a}, \mathfrak{b}, \mathfrak{c}, \mathfrak{d}, \mathfrak{k}, \mathfrak{l},$ e, x, y, z, h) sind bestimmt durch 12-tupel von Daten, die folgende Bedeutung haben: $\mathfrak{p}^{ist}$, $\mathfrak{a}, \mathfrak{b}, \mathfrak{c}, \mathfrak{d}, \mathfrak{k}$ und $\mathfrak{l}$ sind (g+1)-dimensionale Vektoren mit je einer Komponente für jedes Element aus L_0; e, x, y und z sind Boolesche Größen, h ist eine Zahlengröße. $\mathfrak{a}, \mathfrak{b}, \mathfrak{c}$ und $\mathfrak{d}$ dienen der Formalisierung der Bedingungen c_1 - c_4 aus II.1.c, $\mathfrak{l}$ registriert die Einführung der Zuordnungen aus L ins Lehrprogramm, $\mathfrak{k}$ registriert deren Auftreten in den einzelnen Arbeitstakten. Ohne jeweils die methodische oder didaktische Bedeutung im einzelnen zu erläutern, wird nun das Werteverhalten der einzelnen Daten beschrieben: (vgl. die Ausführungen in Graf 1969)

1. h (t) ist die Anzahl der im Zeittakt t bereits in den Lehralgorithmus eingeführten Zuordnungen aus L.
2. x (t) = 0 bedeutet, daß die Eingabe in P zur Zeit t in Lehrquantenform erfolgte, im nächsten Takt hat sie in Frageform zu geschehen.
3. y (t) = 0 solange h (t) $\neq$ g.
4. z (t) = 0 solange für mindestens ein i aus L gilt: $p_i^{ist} < p_i^{soll}$.
5. e (t) = 1, falls die Komponente i (t) von x_{PYM} (t) den Wert 1 hat.
6. k_i (t) = 1, falls r (t) = i; (r (t) aus x_{PYM} (t)).
7. l_i (t) = 0, solange i aus L noch nicht im Lehrprogramm aufgetreten ist.
8. a_i (t) = 0, falls r (t) oder r (t-1) oder r (t-2) = i.
9. b_i (t) = 0, falls $q_i - f_i \geq 2$ (vgl. c_2 aus II.1.c).
10. c_i (t) = 0, falls $q_i - f_i \leqq 0$.
11. d_i (t) = 0, falls $p_i^{ist}/p_i^{soll} \geqq C$ (t) (vgl. c_3 aus II. 1.c).
12. p_i^{ist} (t) ist die Wahrscheinlichkeit, mit der die Zuordnung i zur Zeit t in PYM gespeichert ist.

d) PYM ist ein initialer Automat, der Anfangszustand zur Zeit $t = 0$ ist wie folgt gegeben:
$h(0) = 0$, $x(0) = 1$, $y(0) = z(0) = 0$, $e(0) = 0$,
$\mathfrak{k}(0) = \mathfrak{l}(0) = \mathfrak{c}(0) = \mathfrak{v}$ (Nullvektor),
$\mathfrak{a}(0) = \mathfrak{b}(0) = \mathfrak{d}(0) = \mathfrak{n}$ (Einheitsvektor),
$\mathfrak{p}^{ist}(0) = \mathfrak{p}^{0}$, der vorgegebene Vektor der Anfangs-Speicherwahrscheinlichkeit, vgl. IV. 1.

e) Überführungsfunktion: $z_{PYM}(t) = \delta(z_{PYM}(t-1), x_{PYM}(t))$
(Die Komponenten mit dem Index 0 sind für die Erstellung des Lehrprogramms bedeutungslos, sie erleichtern nur die Darstellung durch Vermeidung von Fallunterscheidungen).

$k_i(t) = 1$, falls $r(t) = i$; $= 0$ sonst, für alle i aus L_0

$l_i(t) = l_i(t-1) + k_i(t)$

$a_i(t) = \bar{k}_i(t) \cdot \bar{k}_i(t-1) \cdot \bar{k}_i(t-2) \cdot \bar{k}_i(t-3)$

$b_i(t) = \bar{k}_i(t) \cdot b_i(t-1) + k_i(t) \cdot (\bar{x}(t-1) + \bar{c}_i(t-1))$

$c_i(t) = \bar{k}_i(t) \cdot c_i(t-1) + k_i(t) \cdot (x(t-1) + \bar{b}_i(t-1))$

$p_i^{ist}(t) = f(\ldots)$ für $r(t) = i$ (vgl. II. 1. c); $= p_i^{ist}(t-1)$ sonst.

$d_i(t) = 0$ falls $p_i^{ist}(t)/p_i^{soll}\ \ C(t)$; $= 1$ sonst.

$e(t) = i(t)$

$x(t) = \bar{x}(t-1)$

$$y(t) = \prod_{i=1}^{9} l_i(t)$$

$$z(t) = y(t) \cdot \prod_{i=1}^{9} \bar{d}_i(t)$$

$h(t) = h(t-1) + 1$ falls $i(t) = 1$; $= h(t-1)$ sonst.

Bemerkung: Die Darstellung der Überführungsfunktion ist nicht ganz korrekt, weil rechts des Gleichheitszeichens z. T. Zustandsgrößen zur Zeit t vorkommen. Es ist aber leicht zu sehen, wie diese Größen zu ersetzen wären. (Für die gewählte Form spricht die größere Übersichtlichkeit.)
Ebenso könnte man den an sich nicht zulässigen Bezug auf die Werte $k_i(t-2)$ und $k_i(t-3)$ durch Einführung eines dreistelligen Speichers vermeiden.

f) Als Ergebnisfunktion haben wir $y_{PYM}(t) = MOD$!

IV.3 Das Lehrsystem als initialer abstrakter Automat Der Moderator (MOD)

Wir haben nun zu zeigen, welches System die in PYM eingehende Stellgröße x_{PYM} erzeugen kann. In den Kapiteln II und III wurde schon ausgeführt, daß es sich dabei um ein System von Teilautomaten handelt, die wir zunächst einzeln zu beschreiben haben. Wir beginnen mit dem Moderator.

Es sei nochmals erwähnt, daß wir, um die Ein- und Ausgabealphabete nicht zu kompliziert zu gestalten, davon ausgehen, daß allen Teilautomaten alle aktuellen Zustände aller anderen Teilautomaten bekannt bzw. zugänglich sind. Für die Rechnerautomaten RWZ usw. läßt sich das über gemeinsame Speicher ohne weiteres regeln, für die Autorsysteme sollen die Zustandsdaten durch Abfrage zugänglich sein. Dies gilt auch für den Inhalt der Zwischenspeicher AUSr und AUSä, die die jeweils aktuelle Zuordnung und die ästhetische Form enthalten.

Wir versehen alle Ein- und Ausgabebuchstaben mit einer ersten Komponente, die festlegt, welcher Automat angesprochen wird; die Herkunft brauchen wir nicht eigens zu codieren, wenn wir voraussetzen, daß in jedem Zeittakt nur Eingabebuchstaben angeboten werden, die im Sinne des Ablaufplans von Kap. III zulässig sind. Wir schließen also fehlerhafte und nicht vom Moderator aufgerufene Eingaben aus.

Manche Ein- oder Ausgabebuchstaben bestehen aus mehreren Anteilen, die an verschiedene Empfänger gehen bzw. von verschiedenen Sendern kommen. Da jeder Ausgabebuchstabe (mit Ausnahme der von MOD) einen Anteil enthält, der sich an MOD richtet, finden auch alle Eingaben an andere Teilautomaten als MOD im richtigen Takt statt. Denn wie vereinbart beginnt innerhalb des Reglersystems mit jeder Eingabe in MOD ein neuer Takt für alle Teilautomaten. Automaten, die innerhalb eines Taktes gar nicht angesprochen werden, bleiben in ihrem Zustand unverändert.

a) Eingabebuchstaben x_{MOD} = (MOD!, {j = 0}) mit j : : = 0/1.

b) Ausgabebuchstaben y_{MOD} = (aut!,{aufruf = 0})

mit aut: : = { PYM/RWZ/APZ/AWZ/RPZ/AMO/RWA/AWA/APA/RPA},

aufruf : : = {gib zuordnungs-nummer/ gib lehrquant/
gib frage/ gib antwort/ ok?/ selbst?/
noch ein vorschlag?/ trotzdem? }

c) Zustände z_{MOD} = (PYM, RWZ, APZ, AWZ, RPZ, AMO, RWA, AWA, APA, RPA, R, E, H, S, T, U, W, ϑ')

sind 18-tupel von Daten. Dabei nimmt jeweils eine der ersten zehn (Booleschen) Komponenten den Wert 1 an, wenn durch die gleichzeitig erfolgende Ausgabe der dem Namen der Komponente entsprechende Teilautomat angesprochen wird, also MOD im nächsten Takt zugänglich für eine Eingabe dieses Automaten wird.

Auch E, R, H, S, R, U und W sind Boolesche Größen; im einzelnen bedeutet:

E = 1: von AWZ wurde Vorschlag auf "Ende des Lehrprogramms" gemacht,
H = 1: Letzter Takt des Reglers,
R = 1: RWZ hat keine Alternative mehr für eine Zuordnung,
S = 1: Als Ausgabe wird die Frage "selbst?" an AMO gerichtet,
T = 1: Als Ausgabe wird die Frage "trotzdem?" an AMO gestellt,
U = 1: Auswahl einer ästhetischen Frageform wurde vorgenommen, Auswahl einer Antwort muß unmittelbar folgen,
W = 1: Als Ausgabe wird die Frage "noch ein vorschlag?" gestellt.

ϑ' ist ein g-dimensionaler Vektor, dessen Komponenten für jede Zuordnung angeben, ob $p^{ist} \geq p^{soll}$; dann ist $d' = 1$; sonst = 0.

d) MOD ist ein initialer Automat, der Anfangszustand hat die Gestalt (1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ϑ').

e) Überführungsfunktion: $z_{MOD}(t) = \delta(z_{MOD}(t-1), x_{MOD}(t))$.

$RWZ(t) = PYM(t-1) + AMO(t-1) \cdot \bar{S}(t-1) \cdot W(t-1) \cdot j(t-1)$
(Im folgenden lassen wir die Zeitwerte weg, alle Größen links vom Gleichheitszeichen werden für t bestimmt, alle Größen rechts beziehen sich auf t-1).

$AWZ = AMO \cdot \bar{S} \cdot W \cdot \bar{j} + APZ \cdot R \cdot \bar{j}$
$RPZ = AWZ$
$APA = RWZ$
$AMO = RPZ + APZ \cdot (\bar{R} + j), + AMO \cdot \bar{S} \cdot \bar{W} \cdot T$
$RWA = AMO \cdot S \cdot \bar{j} + (RPA + APA) \cdot \bar{x} \cdot \bar{U} \cdot j$
$AWA = AMO \cdot S \cdot j + (RPA + APA) \cdot \bar{j}$
$RPA = AWA$
$APA = RWA$
$PYM = (RPA + APA) \cdot (U + x) \cdot j$
$E = AWZ \cdot \bar{j} + \overline{PYM} \cdot E$
$H = APZ \cdot x \cdot z \cdot j + RPZ \cdot E \cdot j + H + AMO \cdot \bar{S} \cdot T \cdot \bar{W} \cdot E \cdot j$
$R = RWZ \cdot \bar{j} + \overline{PYM} \cdot R$

$S = (APZ + RPZ) \cdot j + AMO \cdot \bar{S} \cdot T \cdot \bar{W} \cdot j$
$T = RPZ \cdot \bar{j}$
$U = (RPA + APA) \cdot \bar{U} \cdot \bar{x} \cdot j$
$W = APZ \cdot \bar{j}$
$d_i^*(t) = 1$ falls $p_i^{ist}(t-1) = p_i^{soll}$ (für alle i aus L).
$STOP = PYM \cdot H.$

Die Überführungsfunktion des Moderators läßt sich in Abb. 5 graphisch verfolgen.

f) Ergebnisfunktion: $y_{MOD}(t) = \lambda(z_{MOD}(t-1), x_{MOD}(t))$.

aut (t) = RWZ falls RWZ (t) = 1
= AWZ falls AWZ (t) = 1
usf. ...
= PYM falls PYM (t) = 1.

aufruf (t) = "gib zuordnungsnummer"	falls AWZ (t) = 1
= "gib lehrquant"	falls AWA (t) $\cdot$ x (t-1) = 1
= "gib frage"	falls AWA (t) $\cdot \bar{x}$ (t-1) $\bar{U}$ (t) = 1
= "gib antwort"	falls AWA (t) $\cdot \bar{x}$ (t-1) U(t) = 1
= "ok"?"	falls APA (t) + APZ (t) = 1
= "selbst?"	falls AMO (t) $\cdot$ S (t) = 1
= "noch ein Vorschlag?"	falls AMO (t) $\cdot$ W (t) = 1
= "trotzdem?"	falls AMO (t) $\cdot$ T (t) = 1
= $\bar{O}$	sonst

Es erscheint an dieser Stelle angebracht, eine anschauliche Beschreibung der Ablaufmöglichkeiten des Autor-Rechner-Dialogs einzufügen, um das Verständnis der Automatenfunktionen zu erleichtern. Dabei greifen wir z.T. bereits auf die Ein- und Ausgabebuchstaben der noch nicht behandelten Automaten vor.

Der Dialog beginnt mit einem Anstoß von RWZ. I.a. wird die Zuordnungsnummer 1 vorgeschlagen und APZ wird Zustimmung ergeben (j = 1). Da gewiß noch xz = 0 gilt, erfolgt die Anfrage "selbst?" an AMO. Zustimmung (j = 1) bedeutet, daß der Autor selbst die passende ästhetische Form generieren will, also tritt AWA in Funktion; wir erreichen Punkt 8 in Abb. 5. Dabei erfolgt von MOD der Aufruf "gib lehrquant" bzw. "gib frage" bzw. "gib antwort".

Ablehnung des "selbst?" würde zu RWA und Punkt 6 führen.

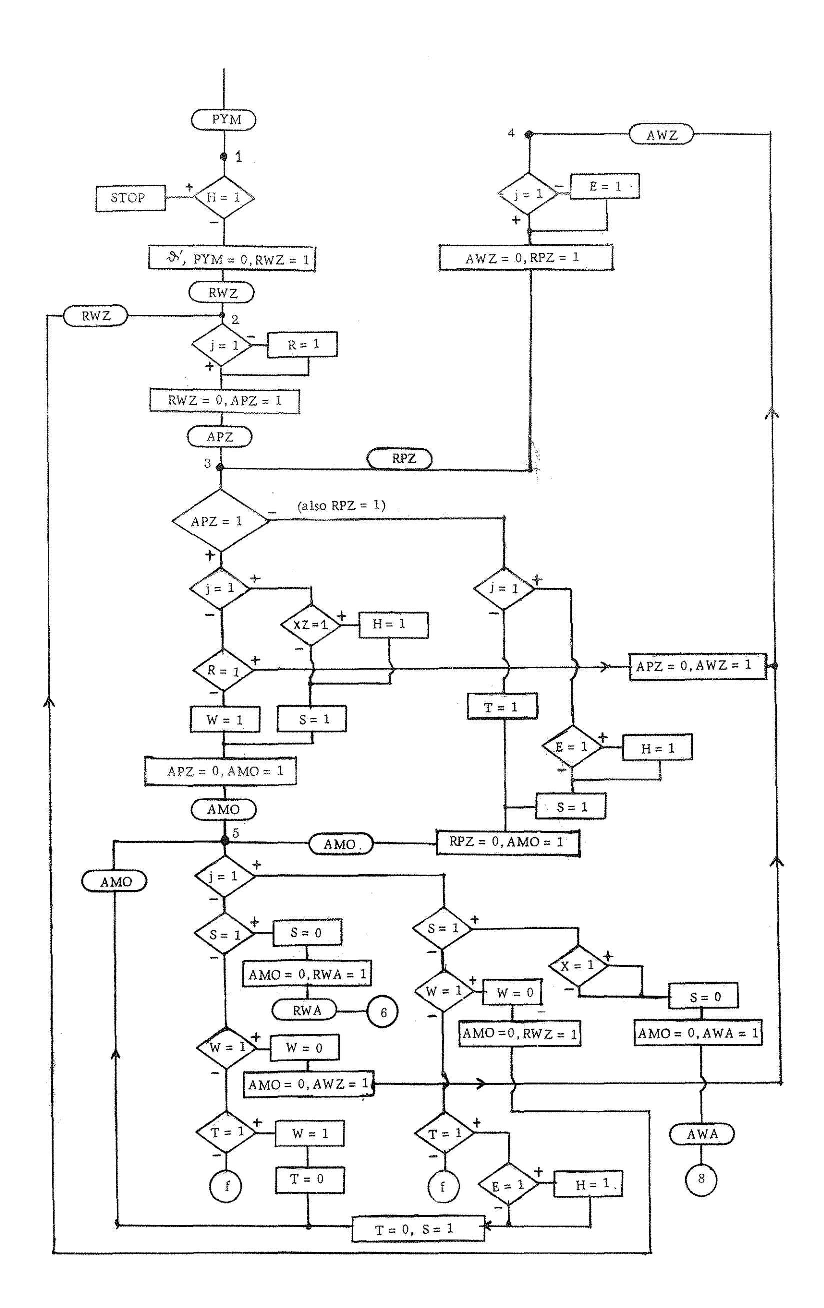


Abb. 5 Zur Überführungsfunktion des Moderators

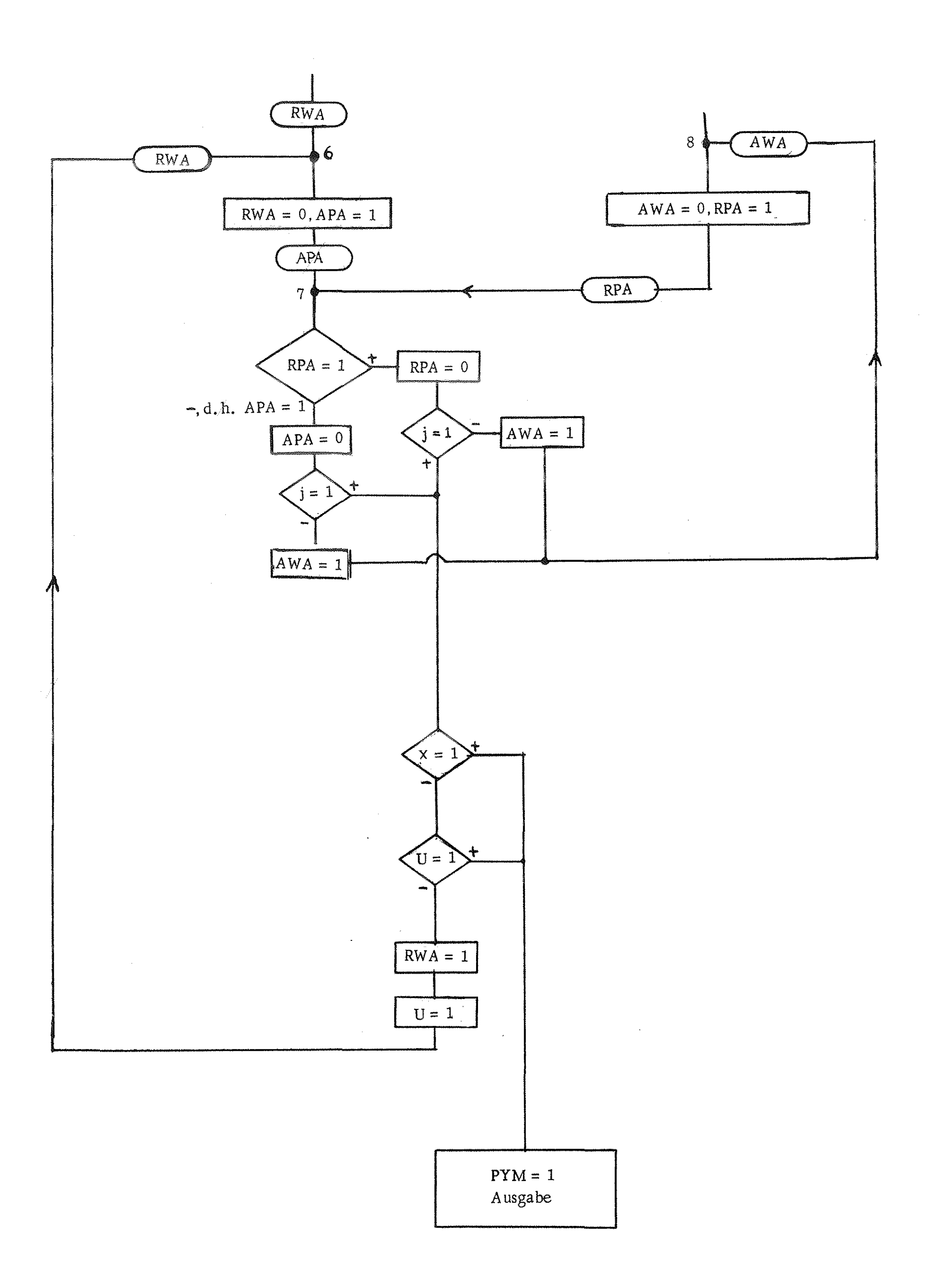


Abb. 5, Teil 2: Zur Überführungsfunktion des Moderators

Auf AWA folgt RPA. Ist die eingegebene Form zu lang (j = 0), so wird AWA gleich wieder aufgerufen, mit der Mitteilung "zu lang". Wenn nicht, so ist im Fall x=1 ein Lehrquant fertig generiert und kann an PYM ausgegeben werden. Bei x = 0 ist zur Frageform gleich noch die Antwort bereitzustellen. Dies wird auf jeden Fall zunächst an RWA übertragen, wobei der Merker U = 1 gesetzt wird. Sobald die Antwort feststeht, u. U. nach einem Umweg über AWA, erfolgt dann wegen $\bar{x}U =$ 1 wieder die Ausgabe an PYM.

Erzeugt RWA die ästhetische Form, so folgt durch APA die Prüfung durch den Autor, angefordert durch "ok?". Lehnt dieser ab (j = 0), so wird er sofort durch "gib lehrquant" bzw. "gib frage" bzw. "gib antwort" zu eigener Eingabe aufgefordert. Das ist u. a. deswegen erforderlich, weil der Rechner gar keine Alternative zur ästhetischen Form anbieten kann. Die Auswahl der ästhetischen Form durch den Rechner wird am Ende von IV. 5 erläutert.

Wir gehen nochmals zu Punkt 2 in Abb. 5. Der Merker R = 1 wird gesetzt, wenn RWZ keine Alternative für die Wahl einer Zuordnung mehr hat. Lehnt der Autor als APZ in diesem Falle, der mit "letzter möglicher vorschlag" angekündigt wird, ab, so wird er sofort als AWZ aufgerufen mit "gib zuordnungsnummer"; man gelangt nach Punkt 4. Ist dagegen R = 0, so erfolgt bei Ablehnung durch den Autor an AMO die Anfrage "noch ein vorschlag?" und man gelangt nach 5. Bei Zustimmung erfolgt wie eingangs erläutert die Frage "selbst?" und damit der Übergang zur Auffindung der ästhetischen Form. Falls xz = 1, also das Lehrziel erreicht ist, wird vorher der Endemerker H = 1 gesetzt. Es ist dann nur noch ein Schlußlehrschritt zu erzeugen.

Von Punkt 4 ausgehend wird RPZ aufgerufen, also eine Prüfung einer vom Autor vorgeschlagenen Zuordnung veranlaßt. Wird "Ende" beantragt, so wird der Merker E = 1 gesetzt. Das Ergebnis der Prüfung sind Mitteilungen über alle möglichen formalen Einwände an den Autor (vgl. Ausgabebuchstaben von RPZ). Über Punkt 3 wird nun, wenn Einwände bestehen (j = 0), die Anfrage "trotzdem?" an AMO erreicht. Der Autor kann also entscheiden, ob er trotzdem mit der von ihm vorgesehenen Zuordnung arbeiten will. Sind keine Einwände vorhanden, so lautet die Anfrage sogleich "selbst?".

Geht man nochmals nach Punkt 5, so ist noch zu behandeln, daß die Anfrage "noch ein vorschlag?" mit nein beantwortet wurde. Dann wird AWZ angesprochen und man gelangt nach 4. Wurde "trotzdem?" mit nein beantwortet, so wird AMO nochmals mit "noch ein vorschlag?" angesprochen, also Rückkehr nach 5. Wird "noch ein vorschlag?" von AMO bejaht, so folgt eine Aktivierung von RWZ und man kommt nach 2.

IV.4 Auswahl einer Zuordnung durch den Rechner (RWZ)

Neben dem Moderator ist der Teilautomat RWZ besonders interessant. Er hat die Aufgabe der Auswahl einer Zuordnung zur Neueinführung oder Wiederholung im Lehrprogramm bzw. für PYM. Wie in Kap. III bereits festgelegt, teilt der MOD diese Aufgabe zu Beginn eines Regeltaktes RWZ jeweils zuerst zu, ehe ggf. AWZ herangezogen wird.

a) Eingabebuchstaben x_{RWZ} = RWZ! (= y_{MOD} mit aut! = RWZ!).

b) Ausgabebuchstaben y_{RWZ} = (MOD!, j)· (APZ!, qu, r, { "neueinführung"=0 } {"letzte notwendige erwähnung" = 0}, {"lehrquanten-überhang" = 0} , { "fragen-überhang" = 0}, { "sperre für abstand verletzt" = 0}, { "sperre für schwelle verletzt" = 0 } , {"letzter möglicher vorschlag" = 0 } , { "soll schon erreicht" = 0 }) · (AUSr!, r, i, f, A, B, D).

Die einzelnen Komponenten der Ausgabebuchstaben werden bei der Darstellung der Ergebnisfunktion erläutert.

c) RWZ hat eigentlich nur die Aufgaben eines Zuordners auszuüben, dürfte also nur einen Zustand haben. Es zeigt sich jedoch, daß RWZ nicht so konstruiert werden kann. Wie bereits erwähnt, sind nämlich Zustände von PYM möglich, die zu bestimmten Zeittakten gar keine Zuordnung als Eingabe zulassen. Diese Schwierigkeit läßt sich nur umgehen, indem für RWZ Zustände eingeführt werden, die die Auswahl eines "näherungsweise" geeigneten Ausgabebuchstabens (als Teil der Stellgröße) gestatten. Konkret kann das heißen, daß man RWZ in die Lage versetzt, eine oder mehrere der Bedingungen von PYM zu vernachlässigen. Dieser Fehler ist dann dadurch auszugleichen, daß die fehlerhafte Zuordnung durch entsprechende Wahl einer ästhetischen Form korrigiert wird, - vorzugsweise durch Eingreifen von AWA -, so daß dann die gesamte Stellgröße doch zum Lernfortschritt in PYM beiträgt.

In unserem Falle führen wir drei Boolesche Größen A, B und D ein. Nehmen diese den Wert 0 an, so werden respektive die Werte der Zustandsvektoren $\mathfrak{a}$, $\mathfrak{b}$ und $\mathfrak{t}$, ϑ von PYM zur Zeit t bei der Zuordnungsauswahl für den Takt t+1 durch RWZ nicht beachtet.

Zusätzlich ist zu beachten, daß RWZ nur Vorschläge für Zuordnungen erzeugt, die noch von APZ überprüft werden. Ggf. wird RWZ im Verlaufe eines Gesamtreglertaktes mehrfach angesprochen. Dies muß auch an verschiedenen Zuständen

von RWZ deutlich werden. Deshalb gehören zu den Zustandsgrößen von RWZ noch zwei Vektoren $\mathfrak{k}'$ und $\mathfrak{w}$ mit je einer Komponente für die g+1 Elemente von L_0. $\mathfrak{k}'$ registriert die jeweils ausgewählte Zuordnung, $\mathfrak{w}$ beschreibt, welche Angebote schon zurückgewiesen wurden.

d) Anfangszustand: A = B = D = 1, $\mathfrak{k}' = \mathfrak{w} = \mathfrak{o}$ (Nullvektor).

e) Überführungsfunktion: $z_{RWZ}(t+1) = \delta(z_{RWZ}(t), x_{RWZ}(t+1))$.

(Alle Größen zur Zeit t, falls nicht anders angegeben).

$$A(t) = e + x(\bar{y} + z) + f_{100} \quad \text{mit} \quad f_{100} = \sum_{i=1}^{9} 1_i a_i (n_i + \bar{x})$$

$$B(t) = e + x(\bar{y} + z) + f_{110} \quad \text{mit} \quad f_{110} = \sum_{i=1}^{9} 1_i a_i (x b_i n_i + \bar{x}(c_i + \bar{n}))$$

$$D(t) = e + x(\bar{y} + z) + f_{111} \quad \text{mit} \quad f_{111} = \sum_{i=1}^{9} 1_i a_i d_i (x b_i n_i + \bar{x}(c_i + \bar{n}_i))$$

$k_i'(t) = 1$ wenn r (t) = i (r aus x_{RWZ} (t)); = 0 sonst

$w_i(t) = \overline{PYM}(t-1)\ (w_i(t-1) + k_i'(t-1))$.

Die Größen A, B und D sind damit so eingestellt, daß RSZ eine für PYM zulässige Zuordnung auffinden kann, und zwar werden dabei die "Sperren" $\mathfrak{a}, \mathfrak{b}, \mathfrak{c}$ und ϑ nur soweit verletzt wie erforderlich. (Reihenfolge: ϑ, $\mathfrak{b}$ und $\mathfrak{c}, \mathfrak{a}$).

Nach dem in Abb. 6 dargestellten Auswahlalgorithmus für eine geeignete Zuordnung ist das genau dann der Fall, wenn die folgende Funktion F den Wert 1 annimmt:

$F = e + f + x(\bar{y} + z)$ mit

$$f = \sum_{i=1}^{9} f_i \quad \text{wobei}$$

$f_i = 1_i (\bar{A} + a_i)(\bar{D} + d_i)(\bar{B} + x b_i + \bar{x} c_i)\, n_i + \bar{x}\bar{n}_i)$, für alle i aus L.

Die Funktionen für A, B und D erhält man durch Einsetzen der Tripel (1, 0, 0), (1, 1, 0) und (1, 1, 1) für (A, B, D) in F.

F = 0 bleibt ausgeschlossen außer evt. im Falle A = B = D = 0. Daß auch letzteres nicht eintreten kann, zeigt Kapitel V.

f) Ergebnisfunktion: $y_{RWZ}(t) = \lambda(z_{RWZ}(t-1), x_{RWZ}(t))$.

1. $j = 0$ falls $\bar{e} \cdot \bar{f}_{111} x\bar{y} = 1$ und $/\{ i \mid \bar{1}_i \bar{w}_i = 1\}/ = 1$

   oder $\bar{e} = 1$ und $/\{ i \mid f_i \bar{w}_i = 1\}/ = 1$

   $= 1$ sonst.

   $j = 0$ bedeutet also, daß RWZ nur noch einen Vorschlag machen kann.

2. qu = "lq" falls $x = 1$, qu = "fr" falls $x = 0$,

   $r(t) = r(t-1)$ falls $em_r = 1$ (Wiederholung einer Neueinführung)

   $= 0$ falls $e\bar{m}_r = 1$ (leere Frage nach Neueinführung)

   $= \min\{ r \mid \bar{1}_r \bar{w}_r = 1\}$ falls $\bar{e} \cdot \bar{f}_{111} x\bar{y} = 1$ (Neueinführung)

   $=$ zufall$\{ r \mid f_r \bar{w}_r = 1\}$ falls $\bar{e} \cdot f = 1$ (Wiederholung)

   $= g^*$ falls $\bar{f}_{111} xyz = 1$ (Schlußlehrschritt).

"neueinführung" falls $\bar{f}_{111} x\bar{y} = 1$

"letzte notwendige erwähnung" falls $p_r^{ist}(t+1) \geq p_r^{soll}$ für $r(t)$.

"lehrquanten-überhang" falls $b_r = 0$

"fragen-überhang" falls $c_r = 0$

"sperre für abstand verletzt" falls $a_r = 0$

"sperre für schwelle verletzt" falls $\bar{d}_r^* \bar{d}_r = 1$

"letzter möglicher vorschlag", falls $j(t) = 0$

"soll schon erreicht" falls $d_r^* = 1$.

3. $r(t)$ wie bei 2.

   $i = 1$ falls $\bar{f}_{111} x\bar{y} = 1$

   $f = 1$ falls $p_r^{ist}(t+1 \geq p_r^{soll}$ für $r(t)$

   $A = a_r$

   $B = xb_r + \bar{x}c_r$

   $D = d_r$

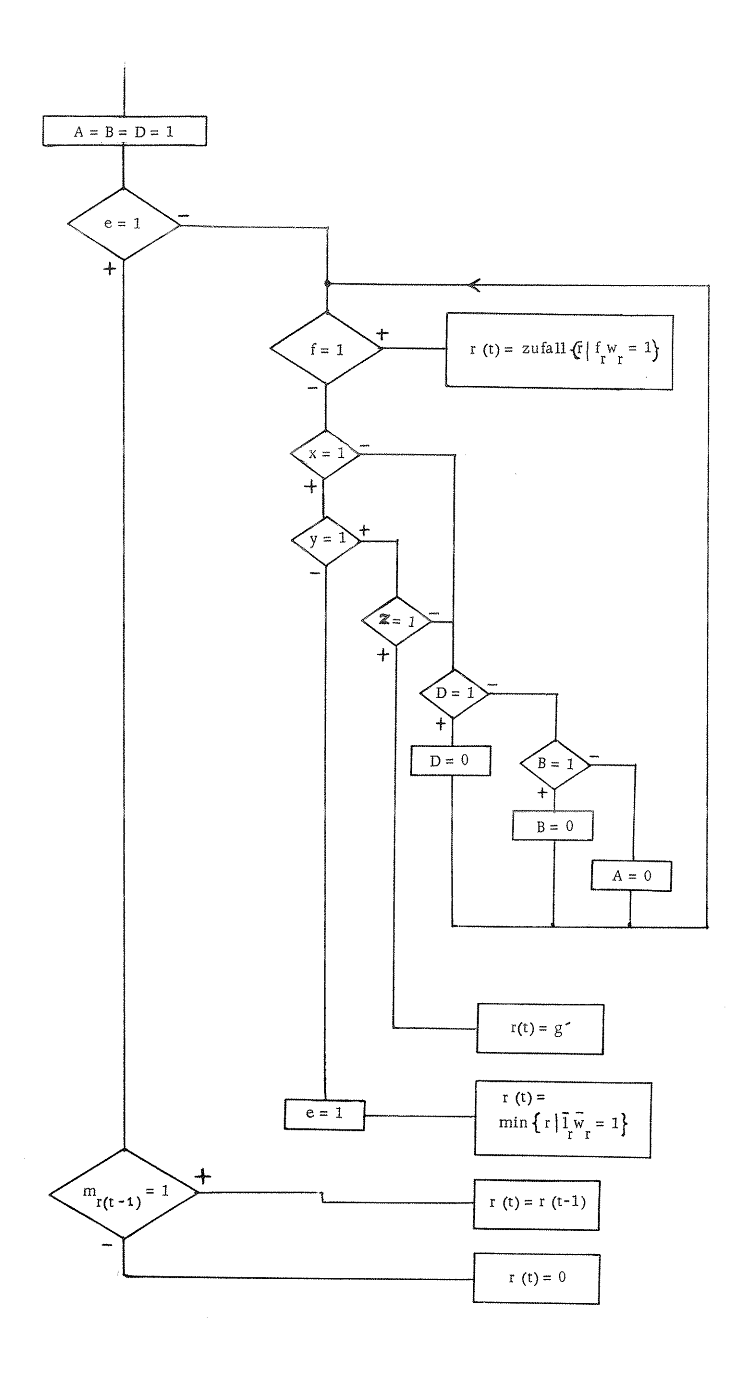


Abb. 6: Zur Ergebnisfunktion von RWZ

Die Strategie von RWZ geht also dahin, zunächst schon im Lehrprogramm eingeführte Zuordnungen zu wiederholen, und zwar unter Einhaltung der Sperren A, B und D. Erfüllen mehrere Zuordnungen die Bedingung für Wiederholung, so erfolgt jeweils unter denjenigen eine Zufallsauswahl, die im Dialog von RWZ noch nicht vorgeschlagen wurden.

Sind keine Wiederholungen notwendig, so wird eine Neueinführung vorgesehen, solange noch y = 0. Dabei wird jeweils die Zuordnung kleinster Rangzahl vorgeschlagen, die noch nicht eingeführt ist und die im Dialog noch nicht vorgeschlagen wurde.

Sind weder reguläre Wiederholungen (A = B = D = 1) noch Neueinführungen möglich, so hebt RWZ sukzessive die Sperren D, B, A auf; wir zeigen in Kapitel V, daß schließlich bei D = B = A = 0 RWZ immer einen Vorschlag erzeugen kann. Über den Merker e wird bei Neueinführungen lediglich geregelt, ob eine Direktwiederholung der Zuordnung als Frage erfolgen soll oder ob eine Pause folgt.

Zusammen mit der Zuordnungsnummer teilt RWZ dem Autor gegebenenfalls besondere Umstände wie "neueinführung", "letzte notwendige erwähnung" u. ä. mit.
Zur Ergebnisfunktion ist noch zu bemerken, daß die Komponenten "letzte notwendige erwähnung" und f sich auf eine Vorausberechnung von p^{ist} beziehen, falls die betreffende Zuordnung in die Stellgröße eingeht. Soweit in diese Berechnung Daten der zugehörigen, an dieser Stelle aber noch nicht bestimmten ästhetischen Form ä eingehen, ist sie noch gar nicht möglich. Man kann sich nur mit einem Durchschnittswert behelfen; allerdings kann dies dazu führen, daß die genannten Komponenten dann falsch bestimmt sind.

IV.5 Auswahl einer ästhetischen Form durch den Rechner(RWA)

Dieser Automat konstruiert mit Hilfe der Mengen G, F und S und der Begriffe der Zuordnungen ästhetische Formen in der Gestalt von Lehrquanten, Fragen und Antworten.

a) Eingabebuchstaben x_{RWA} = RWA!

b) Ausgabebuchstaben y_{RWA} = (MOD!) · (APA!, ä) · (AUSä!ä)

c) RWA ist ein reiner Zuordner mit konstantem Zustand.

d) Ergebnisfunktion: Die in der folgenden Beschreibung verwendeten Größen wie q, a', b', e' usw. beziehen sich auf die Daten der Speicher L, G, F und S. (vgl. IV.1) AUSr enthält r, i, f.

$ä_1 = h_r^{\prime} \cup (t_r^{\prime}, t_r^{\prime\prime})$	falls $\bar{U}\bar{f}x\ (\bar{q}+\bar{i}+\bar{a}^{\prime}) = 1$	(Lq aus Grundform)
$ä_2 = h_r^{\prime} \cup t_r^{\prime}$	falls $\bar{U}\bar{f}\bar{x}\ (\bar{q}+\bar{i}+\bar{a}^{\prime}) = 1$	(Fr aus Grundform)
$ä_3 = g_{b_r^{\prime}}$	falls $x\bar{U}\bar{f}qia^{\prime}n = 1$	(Einführung durch Basaltext aus Lq)
$ä_4 = g_{b_r^{\prime}} \setminus t_r^{\prime\prime}$	falls $\bar{x}\bar{U}\bar{f}qia^{\prime}\bar{n} = 1$	(Einführung durch Basaltext aus Fr)
$ä_5 = h_r^{\prime} \cup (t_r^{\prime}, t_r^{\prime\prime}) \cup s_{s_r^{\prime}}$	falls $\bar{U}fx\ (\bar{q}+\bar{e}^{\prime}(\bar{i}+\bar{a}^{\prime}))=1$	(Lq aus Grundform zum Abschluß)
$ä_6 = s_{s_r^{\prime}} \cup h_r^{\prime} \cup t_r^{\prime}$	falls $\bar{U}f\bar{x}\ (\bar{q}+\bar{e}^{\prime}(\bar{i}+\bar{a}^{\prime}))= 1$	(Fr aus Grundform zum Abschluß)
$ä_7 = g_{b_r^{\prime}} \cup s_{s_r^{\prime}}$	falls $\bar{U}fxq\ (e^{\prime}+ia^{\prime}) = 1$	(Basaltext mit Abschluß)
$ä_8 = s_{s_r^{\prime}} \cup g_{b_r^{\prime}} \setminus t_r^{\prime\prime}$	falls $\bar{U}f\bar{x}q\ (e^{\prime}+ ia^{\prime}) = 1$	(Basaltext als Frage mit Abschluß)
$ä_9 = t_r^{\prime\prime}$	falls $U = 1$	

An einigen Beispielen sei noch ausgeführt, wie die Komponenten der Ergebnisfunktion zu interpretieren sind. Es liege etwa der Moderatorzustand $x = 1$ und $U = 0$ vor, d. h. es ist ein Lehrquant zu generieren und keine Antwort. Es handele sich nicht um die letzte Erwähnung der Zuordnung r ($f_r = 0$). Weiter liege kein Basaltextsatz vor, der r erläutert, oder es handle sich nicht um die Neueinführung von r, oder die Neueinführung von r durch einen Basaltextsatz sei verboten ($q_r = 0$ oder $i_r = 0$ oder $a_r^{\prime} = 0$). Dann wird ein Lehrquant aus einer Grundform $h_r^{\prime}$ und den Begriffen $t_r^{\prime}$, $t_r^{\prime\prime}$ der Zuordnung r zusammengesetzt, $ä_1$.

Sind alle Voraussetzungen wie eben, nur $x = 0$, so wird eine Frage aus einer Grundform $h_r^{\prime}$ und dem ersten Begriff $t_r^{\prime}$ der Zuordnung zusammengesetzt, $ä_2$.

Ist $x = 1$ und $U = 0$, handelt es sich nicht um die letzte Erwähnung von r ($f_r = 0$), existiert ein Basaltextsatz zu r ($q_r = 1$), handelt es sich um eine Neueinführung ($i_r = 1$), ist die Verwendung des Basaltextsatzes am Anfang nicht verboten ($a_r^{\prime} = 1$) und handelt es sich nicht um eine Zuordnung, die nur in Fragen auftreten soll ($n_r = 1$), so wird der Basaltextsatz $g_{b_r^{\prime}}$ als Lehrquant verwendet $ä_3$.

Derselbe Basaltextsatz unter Auslassung von $t_r^{\prime\prime}$ dient als Frage-Einführung, wenn $n_r = 0$, also $ä_4$.

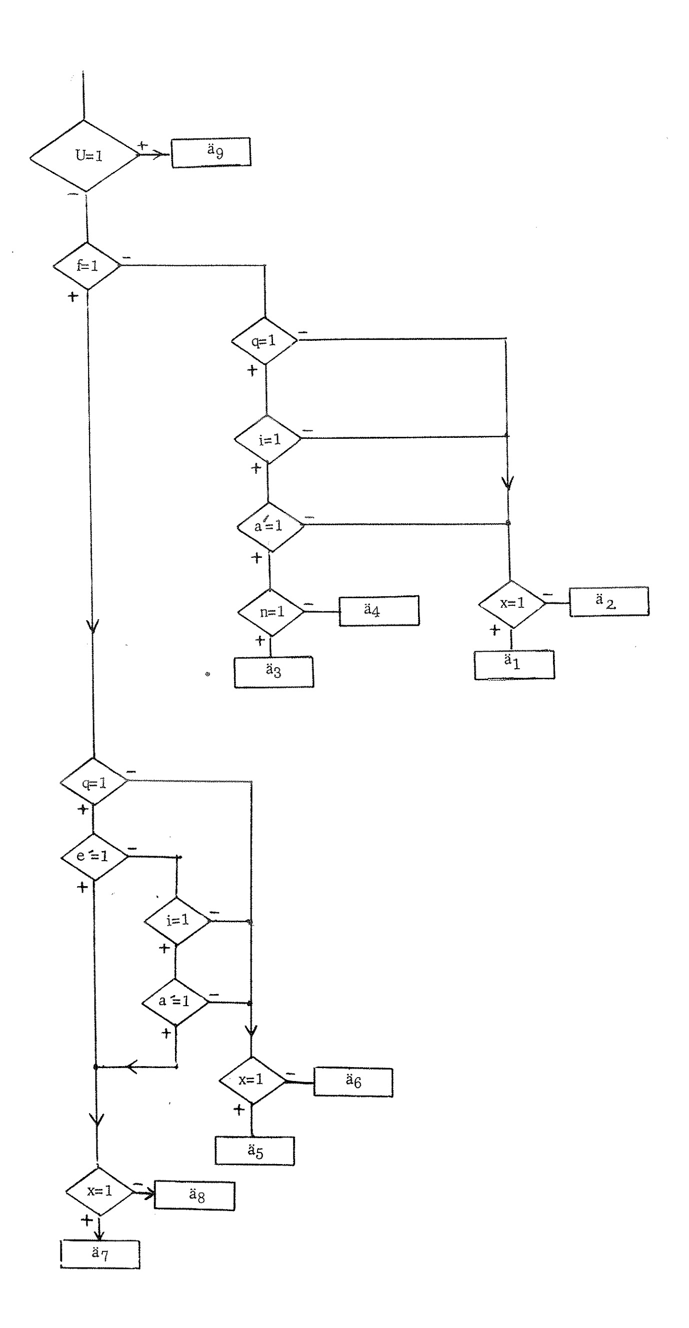


Abb. 7: Zur Ergebnisfunktion von RWA

Mit Ausnahme der Antwort $ä_9$, die jeweils aus dem 2. Begriff einer Zuordnung besteht, entsprechen die übrigen ästhetischen Formen den ersten vier unter Einbeziehung evtl. vorgesehener Schlußbemerkungen.

Die Ergebnisfunktion läßt sich auch in Abb. 7 ablesen.

IV.6 Prüfung einer Zuordnung durch den Rechner (RPZ)

RPZ prüft Vorschläge zur Auswahl einer Zuordnung, die vom Autor (AWZ) in Form einer Zuordnungsnummer kommen. Die Funktion von RPZ besteht sowohl in der Erzeugung des für den Speicher AUSr geeigneten vollständigen Buchstabens als auch in der Information des Autors (AMO) über diesen Buchstaben und das Ergebnis der Prüfung.

a) Eingabebuchstaben x_{RPZ} = (RPZ!) · (RPZ!, r) (von MOD und AWZ)

b) Ausgabebuchstaben y_{RPZ} = (MOD!, j)· (AMO!, {"neueinführung" = 0}, {"letzte notwendige erwähnung" = 0}, {"lehrquanten-überhang" = 0}, {"fragen-überhang" = 0}, {"sperre für abstand verletzt" = 0}, {"sperre für schwelle verletzt" = 0}, {"soll schon erreicht" = 0}, {"zuordnung soll nur gefragt werden" = 0}, {"lehrziel noch nicht erreicht" = 0}, {"vor ende noch eine frage" = 0}) · (AUSr!, r, i, f, A, B, D)

Die einzelnen Komponenten der Ausgabebuchstaben werden bei der Darstellung der Ergebnisfunktion erläutert.

c) RPZ ist ein Zuordner.

d) Ergebnisfunktion:

1. j = 1, falls der AMO-Anteil von y_{RPZ} leer ist (keine Einwände), sonst j = 0.
2. "neueinführung" falls $r \neq 0$ und $1_r = 0$

   "letzte notwendige Erwähnung" falls $p_r^{ist}(t+1) \geqq p_r^{soll}$

   "fragen-überhang" falls $c_r = 0$

   "lehrquanten-überhang" falls $b_r = 0$

   "sperre für abstand verletzt" falls $a_r = 0$

   "sperre für schwelle verletzt" falls $\bar{d}_r \bar{d}'_r = 1$

   "soll schon erreicht" falls $d'_r = 1$

   "zuordnung soll nur gefragt werden" wenn $x\bar{n}_r = 1$

"lehrziel noch nicht erreicht" falls $E\bar{z} = 1$

"vor ende noch eine frage" falls $Exz = 1$

3. $i = 1$ falls $r \neq 0$ und $l_r = 0$

$f = 1$ falls $p_r^{ist} \geqq p_r^{soll}$ (zur Zeit t+1, vgl. S. 36)

$A = a_r$ $B = b_r c_r$ $D = d_r$.

IV.7 Prüfung einer ästhetischen Form durch den Rechner (RPA)

RPA prüft die Zulässigkeit einer von AWA bereitgestellten ästhetischen Form nur hinsichtlich ihrer Zeichenlänge.

a) Eingabebuchstaben x_{RPA} = (RPA !) · (RPA !, ä) (von MOD und AWA).

b) Ausgabebuchstaben y_{RPA} = (MOD!, j) {(AWA !, "zu lang") = 0}

c) RPA ist ein Zuordner.

d) Ergebnisfunktion:
"zu lang" falls /ä/ ≧ 100 und U = 0 oder /ä/ ≧ 30 und U = 1.
j = 0 falls "zu lang" an AWA ergeht, sonst j = 1.

IV.8 Auswahl einer Zuordnung durch den Autor (AWZ)

Hinsichtlich der Teilsysteme des Reglers, die vom Autor realisiert werden, versuchen wir keine vollständige Beschreibung durch einen abstrakten Automaten, betrachten diesen vielmehr als "black box". Das Verhalten des Autors ist durch dessen didaktische Theorie bestimmt, die wir hier nicht zu formalisieren haben.

a) Eingabebuchstaben x_{AWZ} = (AWZ! "gib zuordnungsnummer")

b) Ausgabebuchstaben y_{AWZ} = (MOD !, j) · (RPZ !, r)

c) j = 1 falls r aus L_0, j = 0 falls r = 101 (Endlehrschritt);

IV.9 Auswahl einer ästhetischen Form durch den Autor (AWA)

a) Eingabebuchstaben x_{AWA} = {(AWA !, "zu lang") = 0 } · (AWA !, aufruf)

b) Ausgabebuchstaben y_{AWA} = (MOD !) · (RPA !, ä) · (AUSä !, ä)

c) aufruf:: = {gib lehrquant/gib frage/gib antwort}

IV.10 Prüfung einer Zuordnung durch den Autor. (APZ)

a) x_{APZ} = (APZ!, qu, r, ... (von RWZ)) · (APZ!, "ok?") (von RWZ und MOD)

b) y_{APZ} = (MOD!, j) j ::= {0/1}

IV.11 Prüfung einer ästhetischen Form durch den Autor (APA)

a) x_{APA} = (APA!, ä) · (APA!, "ok?") (von RWA und MOD).

b) y_{APA} = (MOD!, j) j ::= {0/1}

IV.12 Der Autor als Neben-Moderator (AMO)

Der Autor wird vom Moderator jeweils zu Rate gezogen, ehe die Auswahl einer Zuordnung endgültig abgeschlossen wird, insbesondere wenn diese mit Verstößen gegen die Bedingungen des Psychostrukturmodells verbunden ist, (Anfrage "trotzdem?"). Er wird darüber hinaus befragt, ob RAZ weitere Vorschläge machen soll, (Anfrage "noch ein vorschlag?"), und ob er die Formulierung der ästhetischen Form von vornherein selbst übernehmen will, (Anfrage "selbst?").

a) x_{AMO} = (AMO!, { "neueinführung" = 0}.. usw. (von RPZ)).

· (AMO!, aufruf) (von RPZ und MOD)

b) $y_{AMO!}$ = (MOD!, j)

c) j ::= {0/1} aufruf ::= { trotzdem?/noch ein vorschlag?/selbst?}

Nachdem wir nun alle Teilsysteme des Reglers festgelegt haben, können wir diesen formal so beschreiben:

a) Eingabebuchstaben $x_{REG} = y_{PYM}$ = MOD!

b) Ausgabebuchstaben y_{REG} = (PYM!) (r, i, f, ä, A, B, D)

(von MOD und AUS)

c) Zustände sind jeweils die Menge der Zustände aller Teilautomaten, Übergangsfunktion und Ergebnisfunktion sind durch den vom Moderator gesteuerten Dialogablauf bestimmt.

V. Beweis der Lösungsfindung durch den Regler in jedem Takt.

V.1 Die Automaten RWZ und RWA bilden den Regler der Formaldidaktik ALZUDI 2. Diese würde aus der hier dargestellten Dialog-Didaktik dadurch hervorgehen, daß man konstant festsetzt:

y_{APZ} = (MOD!,1), y_{APA} = (MOD!,1),

y_{AMO} = (MOD!,1) bei $(x_{AMO})_{MOD}$ = (AMO!, "trotzdem") und

y_{AMO} = (MOD!,0) bei $(x_{AMO})_{MOD}$ = (AMO!, "selbst").

(Der Aufruf "noch ein Vorschlag?" würde gar nicht erfolgen). Der Dialog-Ablauf entspräche dann einem Ablauf ohne Intervention des Autors, also gerade dem der Formaldidaktik. RWA ist ein Zuordner, deshalb ist das Auffinden einer Lösung ä in jedem Zeittakt gesichert. Anders ist es bei RWZ. Hier haben wir in IV.4 im Falle A = B = D = 0 noch nicht sichergestellt, daß stets eine Lösung für r (t) existiert. Wir beweisen dazu den allgemeineren Satz:

V.2 Es sei /L/ = g = 4, /$\mathcal{n}$/ ≧ g/2, p_i^{soll} ≦ 0,99 für alle i aus L. (Vgl. IV.1). Dann gilt:

1. In jedem Zeittakt t = 0 liefert RWZ eine Komponente r (t), die im Zeittakt t+1 als Eingabekomponente in PYM dienen kann; d.h. F (t) = 1 für alle t = 0 bis zu einem Endtakt T. (vgl. S. 33).
2. Nach endlich vielen Takten sind alle Elemente r aus L unter den Werten von r (t) vorgekommen; d.h. für jedes r aus L gibt es mindestens ein t mit r (t) = r.
3. Nach endlich vielen Takten nimmt r (t) den Endwert g' an; d.h. es gibt einen Zeittakt T mit r (T) = g'.

Beweis: 1. Wie auf S. 33 erwähnt, erhält r (t) immer dann einen geeigneten Wert wenn gilt: F (t) = 1. Sei nun für ein t F (t) = 0.

a) Dann ist notwendig A (t) = B (t) = D (t) = 0: Denn aus der Annahme D (t) = 1 folgt gemäß Definition dieser Größen auch B (t) = A (t) = 1. Dann ist aber F (t) = D (t) = 1, ein Widerspruch! Also ist jedenfalls D (t) = 0. Aus der Annahme B(t) = 1 folgt A (t) = 1 und dann F (t) = B (t) = 1, ein Widerspruch! Also ist auch B (t) = 0. Sei schließlich A (t) = 1. Dann ist F (t) = A (t) = 1, auch ein Widerspruch. D.h. auch A (t) = 0.

b) Wir haben nun F (t) = e + f + x $(\bar{y}+z)$ = 0 mit $f_r = 1_r(n_r + \bar{x})$ für alle r aus L.
Also gilt auch e = f = x $(\bar{y}+z)$ = 0.
Fall 1: x (t) = 1. Dann ist notwendig $\bar{y}$ + z = 0, also auch y = 1. D.h. 1_r = 1 für alle r aus L. Nun wird $f = \sum_{r=1}^{9} n_r = 0$, also $n_r = 0$ für alle r, ein Widerspruch zur Voraussetzung /$\mathcal{n}$/ ≧ g/2.

Fall 2. $x(t) = 0$. Dann ist $f = \sum_{r=1}^{g} 1_r = 0$, also $1_r = 0$ für alle r aus L. D.h. aber $t = 0$ und $\bar{f}_{111} x\bar{y} = 1$, ein Widerspruch zur Ergebnisfunktion.

c) Damit ist die 1. Behauptung bereits bewiesen. Wir wollen gleich noch zeigen, daß der Zustand $A = 0$ für RWZ verhältnismäßig selten angenommen wird, der entsprechende Verstoß gegen die Didaktik also nicht oft eintritt. Sei also $D(t) = B(t) = 0$, aber $A(t) = 1$. Dann ist

$F(t) = e + f + x(\bar{y} + z)$ mit $f = \sum_{r=1}^{g} 1_r a_r (n_r + x)$. Damit diese Funktion zu Null wird, ist notwendig wieder $e = x(\bar{y} + z) = f = 0$.

Fall 1: $x(t) = 1$. Dann ist $y = 1$, also $f = \sum_{r=1}^{g} a_r n_r = 0$.

Aufgrund der Überführungsfunktion von PYM gibt es höchstens drei r mit $a_r = 0$ zur Zeit t. D.h. notwendig $n_r = 0$ für mindestens g-3 Elemente aus L. Wegen $/w/ \geqq g/2$ kann dieser Fall nur eintreten bei $g \leqq 6$.

Fall 2: $x(t) = 0$: Dann ist $f = \sum_{r=1}^{g} 1_r a_r = 0$. Da $a_r = 0$ für höchstens drei Elemente r ist, folgt nun notwendig $1_r = 0$ für g - 3 Elemente aus L. D.h. dieser Fall tritt höchstens ein, wenn noch nicht mehr als 3 Elemente in den Lehralgorithmus eingeführt sind.

2. a) Wir zeigen zunächst, daß die Zuordnung $r = 1$ in das Lehrprogramm eingeführt wird.

Zur Zeit $t = 0$ gilt aufgrund des Anfangszustands von PYM: $\bar{f}_{111} x\bar{y} = 1$. Dann ist nach der Ergebnisfunktion für r (t) (S. 34) $r(0) = 1$.

b) Seien nun schon die Zuordnungen $1, 2, \ldots, r'$ eingeführt und $r' = g$, also $y = 0$. Wir nehmen an, $r' + 1$ würde nie eingeführt. Nun ist notwendig stets $\bar{f}_{111} x = 0$, sonst wäre $r(t) = r' + 1$ laut Ergebnisfunktion. D.h. nun

$$f_{111} = 1 \text{ sooft } x(t) = 1, \text{ also } f_{111} = \sum_{i=1}^{g} 1_i a_i d_i b_i n_i = 1 \text{ für alle } x(t) = 1.$$

Die Zuwachsfunktion für p_i^{ist} ist monoton steigend und stetig für alle i aus L. Es gibt also eine Zahl w, so daß für alle i aus L gilt: $p_i^{ist} \geqq 0{,}99$, falls i w-mal als Wert von r (t) aufgetreten ist. Außerdem wissen wir, daß der Wert $d_i(t) = 0$ für alle nachfolgenden Zeittakte unverändert bleibt, solange $r' = h(t)$ sich nicht ändert. Nach hw Zeittakten gilt nun mindestens für ein r aus L: $d_r = 0$, nach $h^2 w$ Schritten gilt $d_r = 0$ für alle r aus $\{1, 2, \ldots, r'\}$. Dann folgt aber $\bar{f}_{111} = 1$ zu einem Zeittakt t mit $x(t) = 1$, ein Widerspruch zur obigen Ableitung. Also wird $r' + 1$ doch eingeführt. Damit ist durch vollständige Induktion die 2. Behauptung bewiesen.

3. Angenommen, das Lehrprogramm habe kein Ende. Dann darf zu keiner Zeit die letzte Bedingung der Ergebnisfunktion für r (t) gelten, (S. 34), also ist stets $\bar{f}_{111}xyz = 0$. Aus Teil 2 des Beweises wissen wir bereits, daß von einem bestimmten Takt t_g an gilt: $y = 1$, also $l_r = 1$ für alle r aus L. D.h. wir haben insbesondere für die Takte mit $x(t) = 1$ dann stets $\bar{f}_{111}z = 0$, wobei $\bar{f}_{111} = \sum_{r=1}^{g} a_r b_r d_r n_r$.

Spätestens gw Takte nach t_g ist aber mindestens für ein r $d_r = 0$, nach g^2w Takten gilt $d_r = 0$ für alle r. Dann folgt nach Definition von z aber $z = 1$ und außerdem $f_{111} = 0$ für alle Takte mit $x = 1$. D.h. $f_{111}xyz = 1$, ein Widerspruch zur obigen Ableitung. Also ist das Lehrprogramm stets endlich.

VI. Die Sprache für den Autor-Rechner-Dialog. (Bei DIALOG-ALZUDI)

Für den Autor ist die Einsicht in die soeben beschriebene Struktur des Dialog-Verfahrens nicht notwendig. Für ihn genügt die Kenntnis der Bedeutung der sprachlichen "statements", die vom Rechner an ihn ergehen, und der ihm gestatteten Eingabe-"statements". Welche Teilsysteme dabei jeweils in Aktion treten, braucht dem Autor nicht bewußt zu werden.

Wir stellen die Elemente der Dialogsprache abschließend zusammen. Dabei wird nochmals die Funktion des Moderators deutlich. Der Autor braucht keine Aufrufe zu geben; er wird jeweils vom Rechner informiert, was überhaupt als Aktion erforderlich und möglich ist. Die zeitliche Effektivität des Dialogs wird dadurch erhöht, ohne daß die freie Entscheidung des Autors beeinflußt wird. Dieser kann ja die Gestalt des Lehrprogramms jeweils aus seiner Rolle als Neben-Moderator heraus bestimmen.

a) statements des Autors:

(lehrquant) : := "text in bezug zum lehrstoff"
(frage) : := "text in bezug zum lehrstoff"
(antwort) : := "text in bezug zum lehrstoff"
(zuordn. nr): := 0/1/2/3/.../100/101 (ende)
(urteil) : := +/-

b) statements des Rechners (results):

(lehrquant) : := "text in bezug zum lehrstoff"
(frage) : := "text in bezug zum lehrstoff"
(antwort) : := "text in bezug zum lehrstoff"
(zuordn. nr) : := 0/1/2/3/.../100/101 (ende)
(aufruf) : := gib zuordnungsnummer/gib lehrquant/ gib frage/gib antwort/
(anfrage) : := ok?/selbst?/trotzdem?/noch ein vorschlag?/
(bemerkung) : := lq/fr/neueinführung/letzte notwendige erwähnung/lehrquanten-überhang/fragen-überhang/sperre für abstand verletzt/ sperre für schwelle verletzt/ letzter möglicher vorschlag/ soll schon erreicht/ zu groß/ soll noch nicht erreicht/

Zusammenfassung

Die vorliegende Arbeit ist ein Ansatz zur systemtheoretischen Beschreibung von Verfahren der rechnerunterstützten Lehrprogrammierung. Sie betrachtet dabei diese Verfahren als Regelungsvorgänge an Psychostrukturmodellen und beschreibt die erforderlichen Regler als Systeme von abstrakten Automaten.

Angesichts der personalen Gebundenheit von zahlreichen der erforderlichen Regelmaßnahmen, die auf den noch mangelhaften Möglichkeiten der Objektivierung didaktischer Funktionen beruht, werden solche Reglersysteme vorgeschlagen, die personale und durch Rechner objektivierte Teilsysteme integrieren.

Rechnerunterstütztes Lehrprogrammieren kann auf diese Weise als Dialog zwischen einem Autor und einem Rechner betrachtet werden.

Im einzelnen wird zunächst ein Überblick über die vorhandenen und z.T. auch angewendeten Verfahren der Objektivierung von Lehrerfunktionen gegeben (A.I). Daraus wird die Zweckmäßigkeit von Autor-Rechner-Dialog-Verfahren abgeleitet (A.II) und es wird auf die damit verbundenen theoretischen Probleme hingewiesen (A.III).

Im Abschnitt B, dem Hauptteil der Arbeit, wird eine systemtheoretische Erfassung des Prozesses der Lehrprogrammierung angestrebt. Dazu gehört zunächst die Interpretation des Lehrprogrammierens als Regelungsvorgang, der die Belehrung eines Psychostrukturmodells zum Ziele hat (B.I). Weiter wird der in A.II entwickelte Autor-Rechner-Dialog formalisiert beschrieben (B.II). Die Art der Aufgabenstellung dieses Dialogs, nämlich die Entwicklung eines Lehrprogramms, läßt die Einführung einer ebenfalls objektivierten Dialogleitung durch einen eigenen Automaten, den Moderator, als zweckmäßig erscheinen (B.III).

B.IV zeigt schließlich durch explizite Angabe der abstrakten Automaten des Reglersystems, wie die vorgeschlagenen Modelle in einem Spezialfall durch das Rechnerprogramm-System DIALOG-ALZUDI realisiert werden.

Probleme der Lösungsfindung bei Lehrprogrammierprozessen greift B.V auf und weist speziell die Gültigkeit der im Programmsystem DIALOG-ALZUDI enthaltenen Formaldidaktik ALZUDI 2 nach.

Abschließend wird noch die Dialogsprache zusammengefaßt, die für den Autor bei der Benutzung des Dialog-Systems allein interessant ist (B.VI).

Die Bedeutung der Arbeit dürfte vor allem darin liegen, daß hiermit ein übersichtliches Schema für die Strukturierung von Rechnerprogrammen gegeben wird, die Prozesse der Lehrprogrammierung unterstützen sollen.

Schrifttumsverzeichnis

Arlt, W. — ALSKINDI - eine Formaldidaktik zur automatischen Erzeugung von linearen Lehrprogrammen. In: Rollett u. Weltner (Hsg.): Perspektiven des Programmierten Unterrichts, Wien 1970, 237-240

Becker-Frank, S. — Versuche mit den Lehrprogrammiermethoden ALZUDI und ALSKINDI im Deutschunterricht für Ausländer. In: Rollett und Weltner (Hsg.): Fortschritte und Ergebnisse der Unterrichtstechnologie, München 1971, 138 - 141

Blankertz, H. — Theorien und Modelle der Didaktik. München 1969.

Blischke, H. Hilbig, W. Rüßmann, R. — Die halbalgorithmische Formaldidaktik COGENDI. In: Grundlagenstudien aus Kybernetik und Geisteswissenschaft, 9/4 (1968), 97-110.

Braun, K. — Rechnerunterstütztes Lehrprogrammieren für Fachsprachen - Ansätze und Modelle für die Zielsprache Deutsch. In: Rollett u. Weltner (Hsg.): Fortschritte und Ergebnisse der Unterrichtstechnologie, München 1971, 142-148

Closhen, H. — Empirische Ergebnisse mit einem ALZUDI-2-generierten Schul-Lehrprogramm. In: programmiertes lernen und programmierter unterricht 4 (1969), 162-167

Englert, Frank, Schiefele, Stachowiak — Lexikon der kybernetischen Pädagogik und der programmierten Instruktion, Quickborn 1966

Fischer, H. — Problemlösungsverhalten und der rechnerunterstützte Unterricht. In: Rollett u. Weltner (Hsg.): Fortschritte und Ergebnisse der Unterrichtstechnologie, München 1971, 213-217

Frank, H. — Ansätze zum algorithmischen Lehralgorithmieren. In: Frank, H. (Hsg.): Lehrmaschinen in kyb. und päd. Sicht, IV, Stuttgart und München 1966, 70-112.

Frank, H. Zur Objektivierbarkeit der Didaktik. In: Programmiertes lernen und programmierter unterricht 1, (1967), 1-5

Frank, H. Kybernetische Grundlagen der Pädagogik, Bd. 1 und 2, 2. Aufl. Baden-Baden 1969

Frank, H. Prinzipien der objektivierten Formaldidaktik ALSKINDI. In: Grundlagenstudien aus Kybernetik und Geisteswissenschaft 10/1 (1969 a), 23-28

Frank, H.
Frank-Böhringer, B. Begriff und Funktion der ästhetischen Information in Lehrprogrammen. In: Praxis und Perspektiven des Programmierten Unterrichts II, Quickborn 1967, 87-91

Frank, H.
Graf, K.-D.* ALZUDI - Beispiel einer formalen Didaktik. In: Zeitschrift für erziehungswissenschaftliche Forschung 1, (1967), 27-34

Graf, K.-D. Über die Ausführung einer formalen Didaktik mit einer elektronischen Datenverarbeitungsanlage. Qualifikationsschrift der Pädagogischen Hochschule Berlin vorgelegt (1967)

Graf, K.-D. Rechnergesteuerte Erzeugung von Lehrprogrammen nach ALZUDI 2.
In: Neue Unterrichtspraxis 1969, Hefte 5 und 6, 265-278, 330-340

Graf, K.-D. Eine Sprache für den Lehrprogrammierdialog zwischen Mensch und Rechner. In: Grundlagenstudien aus Kybernetik und Geisteswissenschaft 10/4 (1969),121-128

Graf, K.-D. Programmierter Dialog zur Erzeugung von Lehrprogrammen. In: Zeitschrift für erziehungswissenschaftliche Forschung 4 (1970), 18-36.

Graf, K.-D. Zur Entwicklung einer an didaktischen Problemen orientierten Programmiersprache für Rechner. In: Sprache im Technischen Zeitalter 33, (1970 a), 25-33.

Graf, K.-D.
Illner, H. Rechnererzeugte ästhetische Information und ihre Lernwirksamkeit in einem formaldidaktisch erzeugten Lehrprogramm. In: Grundlagenstudien aus Kybernetik und Geisteswissenschaft 11/4 (1970), 125-136

Hartley, B. Putschkat, F.	Computer assisted learning - Individuelles Lernen und Üben mit einer Datenverarbeitungsanlage. In: Lehnert, U. (Hsg.): siehe dort, 82-91
Heinrich, P.-B. Weltner, K.	Über Erweiterungen der Anwendung der formalen Didaktik ALZUDI. In: Rollett u. Weltner (Hsg.) Perspektiven des Programmierten Unterrichts, Wien 1970, 253-256
Lahn, W.	Eine Methode zum Entwurf von Lehrprogrammen. Manuskript Pädagogische Hochschule Berlin, 1970
Lánský, M.	VERBAL - Entwurf eines Algorithmus zur Bestimmung der optimalen Verteilung von Explanationen im Lehrprogramm. In: Rollett u. Weltner (Hsg.): Perspektiven des Programmierten Unterrichts, Wien 1970, 66-70
Lánský, M.	Weiterentwicklung der Methode VERBAL - Methodische Hinweise für die Vorbereitung des Lehrstoffs. In: Rollett u. Weltner (Hsg.): Forschritte und Ergebnisse der Unterrichtstechnologie, München 1971, 119-125
Lánský, M.	Kybernetisches Modell des Gruppenlernens. In: Rollett u. Weltner (Hsg.): Fortschritte und Ergebnisse der Unterrichtstechnologie, München 1971 a, 246-252
Lehnert, U. (Hsg.)	Elektronische Datenverarbeitung in Schule und Ausbildung. München und Wien 1970
Müller, K. Wolber, G.	COURSEWRITER - eine Programmiersprache zum Schreiben von computerunterstützten Lehrprogrammen. In: Lehnert, U (Hsg.) siehe dort, 48-54
Nees, G.	Programmierter Dialog. In: H. Frank (Hsg.): Lehrmaschinen in kyb. u. päd. Sicht, IV, Stuttgart und München 1966, 155-168
Nicklis, W.	Kybernetik und Erziehungswissenschaft, Braunschweig 1967

Scharmann, Th. Das Motivationsproblem beim Programmierten Gruppenunterricht. In: Rollett und Weltner (Hsg.) Fortschritte und Ergebnisse der Unterrichtstechnolo gie, München 1971, 239-245.

Schulz, W. ALZUDI ist keine Didaktik. In: programmiertes lernen und programmierter unterricht 1, (1967), 130-132.

Stahl, V. Lernen im Dialog mit einem Rechner. In: data report 5 (1970), 8-13

Stobbe, P. LIDIA. In: data report 5 (1970) 14-19

Thomas, W.
Otto, G. Film- und computerunterstützter Gruppenunterrich In: Lehnert, U. (Hsg.) siehe dort, 92-99

Waldau, E. Die Programmierung geographischer Namen und Daten mit Hilfe der Algorithmischen Zuordnungs-didaktik. Wissenschaftliche Facharbeit, Berlin 19

Weltner, K. DIAGRAMM-ALZUDI als Erweiterung des Anwendungsbereiches rechnererzeugter Lehrprogramme. In: Grundlagenstudien aus Kybernetik und Geisteswissenschaft 9, (1968), 111-113

Zielke, W. Stichwortschaubild und Faktenanalyse als Hilfsmittel rationellerer Lernprogrammerstellung. In: Rollett u. Weltner (Hsg.): Fortschritte und Ergebnisse der Unterrichtstechnologie, München 1971, 75-86